An investigation into the use of genetic algorithms for shape recognition

Thesis

An Investigation into the Use of Genetic Algorithms for Shape Recognition

THESIS

Submitted for the Degree

of

DOCTOR of PHILOSOPHY

of the

OPEN UNIVERSITY

by

Thomas Michael Egan
BSc MSc

September 1998

Pertains to the Disciplines of Image Processing and Genetic Algorithms with reference to the Digital Signal and Parallel Processing Disciplines

Abstract

The use of the genetic algorithm for shape recognition has been investigated in relation to features along a shape boundary contour. Various methods for encoding chromosomes were investigated, the most successful of which led to the development of a new technique to input normalised 'perceptually important point' features from the contour into a genetic algorithm. Chromosomes evolve with genes defining various ways of 'observing' different parts of the contour. The normalisation process provides the capability for multi-scale spatial frequency filtering and fine/coarse resolution of the contour features. A standard genetic algorithm was chosen for this investigation because its performance can be analysed by applying schema analysis to the genes. A new method for measurement of gene diversity has been developed. It is shown that this diversity measure can be used to direct the genetic algorithm parameters to evolve a number of 'good' chromosomes. In this way a variety of sections along the contour can be observed. A new and effective recognition technique has been developed which makes use of these 'good' chromosomes and the same fitness calculation as used in the genetic algorithm. Correct recognition can be achieved by selecting chromosomes and adjusting two thresholds, the values of which are found not to be critical. Difficulties associated with the calculation of a shape's fitness were analysed and the structure of the genes in the chromosome investigated using schema and epistatic analysis. It was shown that the behaviour of the genetic algorithm is compatible with the schema theorem of J. H. Holland. Reasons are given to explain the minimum value for the mutation probability that is required for the evolution of a number of 'good' chromosomes. Suggestions for future research are made and, in particular, it is recommended that the convergence properties of the standard genetic algorithm be investigated.

Acknowledgements

I would like to thank Steve Holloway and Dr Tony Moy for their help during the preparatory stages of this research work. I wish to convey to my supervisors Dr Nick Hallam, Dr Phil Picton and Dr Adrian Hopgood the appreciation that I have for their very helpful discussions about the research, much needed proof reading and considerable encouragement over the years. I would also like to acknowledge the support that my wife Rosemary and children John and Kerri gave me throughout the project. Many thanks also to my colleague Dr Nory Nakahee for his many presentation corrections and content suggestions. Finally to my wife Rosemary, appreciation for the excellent work she has done checking and formatting this thesis.

Contents

5. Contour Shape Observation Using Chromosome Encoding

6. Contour Shape Recognition Using Chromosome Encoding

7. Chromosome Fitness Calculation

8. Schema and Fitness Analysis

9. Summary and Conclusions

Figures

1. Introduction

1.1 Background to Research Work

This thesis presents the research work performed to investigate the use of genetic algorithms for recognising shapes in a digitised image. Shape recognition is an image processing technique that is hard to achieve with a digital processor, because the instruction set provided on a general purpose processor is not necessarily appropriate for the type of image analysis that can so easily be performed by a human observer.

A genetic algorithm is a programming technique that is modelled on the principles of evolution in nature. Instructions are encoded into a chromosome and are used to calculate a fitness for each chromosome in a population. A number of solutions to a problem are evolved by selection of the fittest chromosomes for breeding. Crossover and mutation operators influence this reproduction process. Genetic algorithms are usually used in optimisation problems where a single 'best' solution is required, and have been used extensively for image processing problems. In these types of image processing problems, the parameters associated with various spatial filters, templates and threshold operators are encoded into the chromosomes and the 'best' set of parameters is evolved. The encoding of the chromosome is, in general, independent of the problem but must be 'matched' to the problem environment for success to be achieved.

The genetic algorithm has been chosen as the subject of this research because it provides a search mechanism that does not readily become influenced by a 'local' solution and therefore can be made to search more widely over the solution space. New solutions may also be discovered that can not be found by other techniques. The genetic algorithm can provide a number of less 'good' solutions, should the 'best' solution be unattainable in a finite time. Because the fitness of each chromosome is independent of the rest of the population, the genetic algorithm fitness calculation is an inherently parallel process. This parallelism may have real-time advantages when applied to the shape recognition process. The standard or canonical genetic algorithm was used in this research because

it lends itself to analysis, to show that it behaves according to the Schema Theory of J. H. Holland (1992). The schema theory does not say anything about the convergence of the genetic algorithm, but techniques are considered that can modify the selection process of the standard genetic algorithm to achieve convergence.

One of the main features of an object in image processing applications is its shape. The shape of an object can be identified by the curve or contour surrounding the outer boundary of the shape. This research has considered alternative methods for encoding the boundary shape into a chromosome. The initial work looked at the possibility of the chromosome position in an image encoding information about a shape. Chromosome motion, following a boundary contour, was also investigated as a way to encode the contour features. Both of these techniques had limitations. The high curvature points along a spatially normalised shape contour were shown to be suitable inputs to a genetic algorithm. The research has designed a novel method for extracting the information about the high curvature points on a contour and a corresponding chromosome that is encoded with instructions on how to observe various parts of the contour, rather than containing parameters for operations directly on the pixels in the image. Extracting the appropriate information about a shape contour in a suitable form for input into a genetic algorithm proved to be a difficult problem and in practice took a long time to solve. The chromosome used in this research encoded different ways to 'observe' the parts of a shape contour rather than the usual method of parameter encoding for filter or model templates.

The research has investigated if the evolution of a single 'best' way to observe part of a contour would be sufficient, in general, for recognition purposes. The evolution of a variety of chromosomes will usually be required, so that various parts of a contour can be observed in different ways and thus different, but similar, shape contours can then be distinguished. This research has also considered the structure of the genetic algorithm that is needed to achieve this

requirement and, in particular, a new measurement technique has been developed that can be used to control the diversity of 'good' solutions. Some applications of genetic algorithms may have to be satisfied with a number of less fit solutions, but this shape recognition genetic algorithm application **must** have a number of less fit solutions to succeed.

The research has also designed a new and effective recognition process that can be used to identify which chromosomes, from the variety evolved by the genetic algorithm, are suitable for recognition of training set shapes in a test set of similar but different shapes. Two thresholds have been identified as control parameters. The value of each threshold is not critical but they have to be adjusted together to give good recognition. The research suggests that the values of these two thresholds could be adjusted by the application of a second concurrent genetic algorithm.

The research has also analysed the standard or canonical genetic algorithm developed for this research work according to the schema theory of J. H. Holland (1992) and has shown that this genetic algorithm behaves according to this theory. An argument is also presented to explain why certain genetic algorithm parameter values produce a number of 'good' solutions rather than just the 'best' solution. The effect of one gene on another in the chromosome (epistasis) is analysed and various methods for performing this type of analysis have been investigated.

1.2 Shape Contour Features

Images have contours that define the various intensity levels and, thus, the various shapes in an image. For a digitised image, the contours define the boundaries of the shapes at the different intensity or grey levels in the image. Recognition of shapes relies heavily on detection of these boundary contours and an analysis of the contours such that similar shapes can be recognised. Classifying the contours into groups requires features (or characteristics) of the contours to be measured so that the various shapes of the contours can be easily identified. Typical features of a

contour that are usually measured include curvature (especially the positions of high curvature), moment descriptors and Fourier descriptors (both described later), and possibly some form of syntactical representation (e.g. convex, very concave, etc.). Moment descriptor calculations provide various relationships between the (x, y) position of a point in the image (pixel) in terms of the x and y co-ordinates raised to the power p and q respectively (where p and q are integers usually in the range 0 to 3).

Fourier descriptors are obtained by calculating the Fourier transform of a contour and defining the lowest spatial frequencies as descriptive of the contour. Syntactic methods require descriptions of the various parts of the contour that are often measured by the curvature of the contour. The features for various shapes have to be collected from a number of examples of the shapes, and usually are referred to as the 'training set'. The recognition of a new shape requires a classification process that searches the feature 'database' for the best match between the features of the new shape and the features in the database. If the features of various shapes 'overlap', i.e. some shapes have similar features, then the recognition of a particular shape can be difficult. A further description of the contour(s) is also usually required to establish recognition of a part of an object. Occlusion of part of a contour should, if possible, be able to be taken into account when matching a contour database to the image.

Generating contours from an image is by no means easy. Noise and low contrast on the image distort the various contours. Applying a threshold to the image at various levels will enable a boundary tracing process, (see also Freeman, 1961 and 1977) to identify the pixel co-ordinates of each of the contours. Figure 1-1 shows a photograph of a car that has been scanned into the computer. Figure 1-2 shows the contours of the same picture for a simple threshold of 150 in a grey-scale of 0 to 255. A boundary/contour tracer process provides the x, y co-ordinates of the number-plate of the car (Figure 1-3 and Figure 1-4). The alphanumeric content of the number-plate can be

Figure 1-1: Scanned Photograph

clearly seen.

Applying band-pass spatial filters, e.g. Wavelets (see Kaiser, 1994 and Meyer, 1993), to the image will produce a version of the image that has a positive and negative contrast. A boundary tracing process, which can identify the 'zero crossings' between the two contrasts, can also provide a shape contour for analysis.

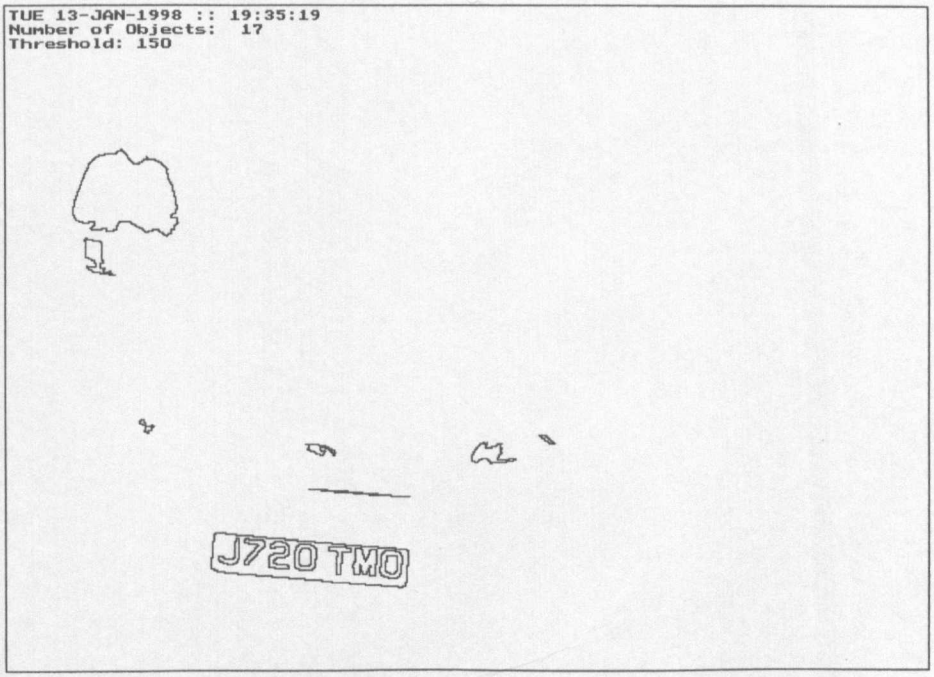

Figure 1-2: Contours at Threshold 150

	Contour Item	Number of Contour Points
1	Box	354
2	J	70
3	7	86
4	2	114
5	0 (Outer)	68
6	0 (Inner)	42
7	T	72
8	M	120
9	O (Outer)	72
10	O (Inner)	42

Figure 1-3: Number-Plate Contour Details

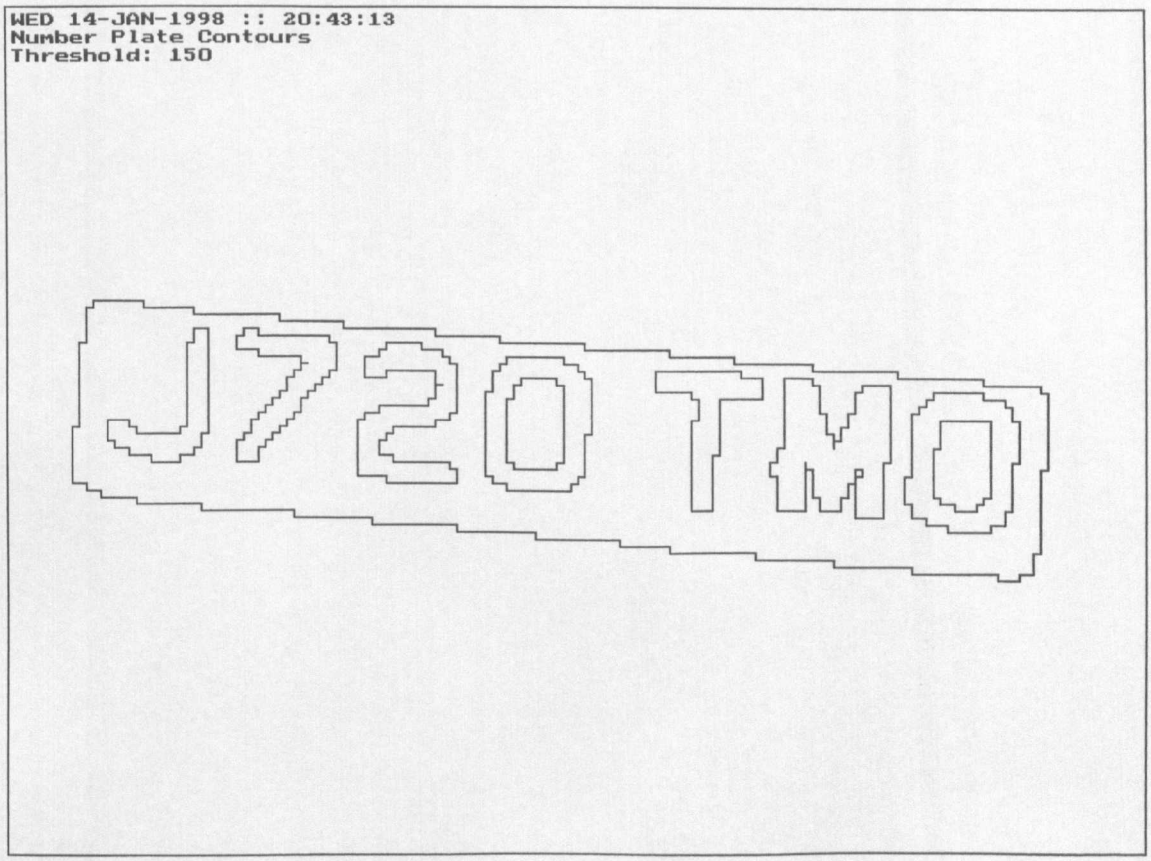

Figure 1-4: Number-Plate Contours

A spatial filter that enhances the edges in the image can also be used together with an edge joining process and some form of morphology (see Sonka, 1994, page 422) to identify the shape contours on the image.

Multi-scale and multi-resolution capability is a useful added feature for a contour recognition algorithm because recognition can be achieved using both coarse and fine detail. Multi-scale and multi-resolution versions of an image, and therefore its contours, can be obtained by filtering the image so that the spatial frequencies at different resolutions are made available for further processing.

In order to improve the shape recognition process, features that are invariant to position, rotation, scale and contour starting point should be used so that variation in the shape that usually occurs, e.g. in a face recognition application can be accommodated.

In order to be able to find any of the contours in an image, a pre-processing operation is usually applied to the image, at various scales if required, which can enhance the appropriate parts of the image to improve the detection of the contours. The pre-processing can be considered to be a filtering operation. The simplest filtering consists of a set of grey level thresholds at which a boundary-tracing algorithm can provide the (x, y) co-ordinates of the contours at each grey level. The next level of filtering provides the first differential of the image grey levels and visually appears as the edges in the image. A further level of filtering calculates the second differential of the image grey levels, and visually also appears as edge information but also includes zero-crossing information.

The zero-crossing information may be used to estimate the main boundary of a shape in the image. The inside of this major boundary will be a 'region' which contains the particular combination of pixels, descriptive of the particular shape. This interior of the shape will also contain contours, which can be used to assist in the recognition process.

Edges are often considered for the possible basis of a shape description. Hough transform methods (see Leavers, 1992) for separating regions and identifying their boundaries, attempt to isolate a region, (e.g. using a series of straight lines) and from there attempt to describe the contents of the region in some way. Edges can be ill defined and fragmented and hence the Hough transform methods may not be able to distinguish the various regions correctly.

Normalising of the measured contour is essential so that the analysis is able to give a consistent description of the contour in comparison to those stored in a database. The normalisation will be for such elements as position, rotation, scale and contour starting point. The training method used to produce a database has to be considered, with the training occurring off-line or on-line as the algorithm is running. Unsupervised learning of the characteristics of a shape can have advantages when tracking various shapes in the image, which are to be recognised on a frame by frame basis.

Moment descriptors and Fourier descriptors do not give a satisfactory description of the contour when used alone. These descriptors provide supporting evidence, but usually fall short of a complete description of the shape. Normalisation for translation, rotation and scale presents its own problems. Distortion, (e.g. skewing) should, if possible, also be considered in the description mechanism. Correlation between moment descriptors and Fourier descriptors may indicate similarity between various shapes. Moment descriptors and Fourier descriptors also remove some of the spatial information from the shape features and can therefore be susceptible to providing similar features for shapes that have similar, but quite distinct, distinguishing spatial differences.

The variations in direction as a contour is traced potentially provide the necessary information for shape recognition. Perceptually Important Points (PIPs) segment a closed contour into the dominant features, which can be used for recognition. Four or eight way connectivity on the contour (see Freeman, 1961) can be used, each highlighting various shape features in different ways. The variation in direction as measured by PIPs can be as fine a grain as required. Possible multi-resolution descriptions may be possible using low pass filtering of the contour in combination with the PIP information. Multi-resolution along the curve can also be obtained by low pass filtering the curve using a variety of cut-off frequencies. A mixture of high and low pass filters can be applied (e.g. Wavelets) and the PIPs for the different scales can be analysed.

A mathematical representation of the contour is in itself not sufficient, and is in some cases ambiguous, without a more general description which groups the directions (in the x, y plane of the image) between the PIPs into some orderly manner. Syntactic methods for describing the PIPs might be possible but seem to rely on a set of basic syntactical connections which are provided by the user. Automatic syntactic descriptions can be difficult to achieve.

1.3 Evolutionary Computation

Fogel (1994) presented a very good summary of "Simulated Evolutionary Optimisation" and noted that these simulations result in stochastic optimisation techniques that can often out-perform classical methods of optimisation when applied to difficult real-world problems. "There are currently three main areas of research in simulated evolution: genetic algorithms, evolutionary strategies and evolutionary programming. Each method emphasises a different facet of natural evolution. Genetic algorithms stress chromosomal operators. Evolutionary strategies emphasise behavioural changes at the level of the individual. Evolutionary programming stresses behavioural change at the level of the species." Fogel (1994) also observed that each of the above methods maintains a population of trial solutions, imposes random changes to those solutions and incorporates the use of selection to determine which solutions to maintain into future generations and which to remove from the pool of trials. "The greatest potential for the application of evolutionary optimisation to real-world problems will come from their implementation on parallel machines, for evolution is an inherently parallel process."

Kinnear (1994, page 5) presented a brief description of the various *Evolutionary Computational* methods. The predominant styles are summarised, from this reference as follows:

1. Evolutionary Strategy defines a style of evolutionary computation frequently associated with engineering optimisation problems. The structures that undergo adaptation are typically sets of real-valued *objective variables* that are associated with real-valued *strategy variables* in an individual. Fitness is determined by executing task specific routines and algorithms using objective variables as parameters. The strategy variables control the way in which mutation varies each objective variable during the production of new individuals. Recombination is usually employed for both objective variables as well as strategy variables.
2. Evolutionary programming operates on a variety of representational structures, frequently real-valued objective variables, although more complex structures have been used (e.g. finite state

machines). Again, the objective variables function as arguments to task specific routines and algorithms designed to solve the problem of interest. Mutation is the sole genetic operator employed, with significant strategy built into the overall algorithm to direct the mutation in a computationally fruitful direction.

3. Genetic algorithms frequently operate on, often binary, fixed length character strings as the structure undergoes adaptation. The fixed length strings are usually referred to as 'chromosomes'. Fitness is determined by executing task specific routines and algorithms using an interpretation of the character string as the set of parameters. Sexual recombination, also called *crossover*, is the principal genetic operator employed, with mutation usually included as an operator of secondary importance. Other representational structures are possible, and are being used more frequently.
4. Genetic programming is an offshoot of genetic algorithms in which the computer structures that undergo adaptation are themselves computer programs. Specialised genetic operators are used that generalise sexual recombination and mutation for the tree-structured computer programs undergoing adaptation.

1.4 Genetic and Evolutionary Algorithm Structure

The basic features of the genetic or evolutionary algorithms are summarised as follows:

1. A population of chromosomes, each of which encodes various properties and/or actions. The chromosomes may be of fixed or variable length. Fixed length chromosomes are easier to deal with, while variable length chromosomes have problems when the length of the chromosome is artificially curtailed. The values stored in the chromosomes may be bit patterns, real numbers or functions that define a particular action to be taken by the cell.
2. An initialisation process which sets values into the chromosomes of each cell in a defined manner. Often the initial values for the chromosomes can be random or seed values that are

appropriate to the problem. These seed values may be produced from measurement made on the environment of the problem.

3. The whole population of cells is activated and their behaviour, under the control of the chromosomes, provides each cell with a particular fitness. The method for calculating the fitness will depend on the problem that is being solved. This process can be performed in parallel and the fitness values can either be transferred to a central point for use in the reproduction process, or populations may compete with each other in a tournament fashion. The definition of the fitness calculation for a particular problem can present problems, but the definition must ensure that fitter cells are in fact going to provide the correct solution to the problem. This point is very important in the case of shape recognition, because some fitness measures may not give the correct shape, as seen by a human observer, as the apparent fitness increases.
4. A reproduction process takes place each generation. The cells are usually selected in proportion to their fitness. The fitness calculation may not produce a set of fitness values for the population that are linear. The fitness vales can be sorted into the correct order and a ranking function can be applied. This ranking function should transform the original non-linear fitness values into a linear set of fitness values. Groups of cells will then have similar fitness values and the probability of choice will therefore depend on a range of values rather than on the individual fitness values. A form of 'elitism' may also be used, where the fittest $n\%$ of the population is automatically transferred to the next generation. The choice of cells for reproduction and the choice of the offspring must be chosen carefully so that premature convergence is not produced.
5. Genetic operators are used after selection to breed new chromosomes from the selected parents. Crossover and mutation are the two most important operators. Both operators have a defined probability of being activated for any two parents. Some parents will be transferred to the next generation without being changed. This property helps to preserve the genetic mix in the

population, and should help to avoid premature convergence. Crossover is usually the main operator, and mixes the chromosomes of the parents in such a manner that improved solutions can be developed as the generations proceed. Mutation, usually with a much lower probability than crossover, is required to try for unusual solutions periodically in order to avoid local minima in the problem space.

6. The contents of the chromosome can be represented in a variety of ways. The traditional genetic algorithm uses bit patterns to represent the problem information. Other representations use real numbers or integers to code the problem information. The chromosome elements can also represent different aspects of the problem, e.g. part of the chromosome may be a condition while the rest of the chromosome may control the action of the cell if the condition is satisfied. The chromosome may also be coded such that one part indicates that a particular function is to be used with the next part of the chromosome, indicating the particular parameters to be used with that function. In this type of representation the whole chromosome can activate more than one function. The classical or standard genetic algorithm requires that the length of the chromosome is fixed, while other evolutionary algorithms allow variable length chromosomes.
7. The new population, which is produced as a result of the application of the genetic operators, is then available for the next generation in 3 above. The whole cycle 3, 4, 5 and 6 is repeated up to a maximum number of generations or until some criteria is satisfied, e.g. when the change in average population fitness is below a certain threshold. The various parameters associated with a genetic algorithm may also be optimised by applying another genetic algorithm to the parameter values, such that a set of populations compete with each other until the best population is developed with the optimum parameters. The genetic algorithm, which is used to develop the best parameters, must be much simpler than the main genetic algorithm because it should have less *a priori* parameters itself.

1.5 Choice of Evolutionary Strategy

The genetic and/or evolutionary approaches have been chosen for this research work because they allow new solutions to be evolved that may not have been discovered before and are approaches that will not, in general, incorrectly find local solutions rather than the best solution. Shape recognition usually requires a large feature database on which to work, and the genetic algorithm allows for this database to be efficiently searched. The genetic reproduction requires that the whole population is joined at that instant, but the calculation of the fitness for each member of the population can be performed in parallel on a network of processors. Thus the training process required to produce the shape recognition feature database can be performed almost entirely in parallel. This property can have 'time' advantages when new shapes are analysed in order to update the training database.

The fitness calculation can be performed in parallel and the collection of the feature data can also be performed in parallel for a number of contours. The population reproduction has to be performed sequentially, but usually is not processor intensive. Some genetic algorithm research reported in the literature has investigated a tournament approach in which a number of separate populations evolve in parallel with periodic crossbreeding.

Difficulties associated with any fitness calculation that involves two-dimensional data are highlighted in the research described in Chapters 3 and 7. Fitness calculations, using the mean and standard deviation for each quantity to estimate the fitness for each variable, can have problems when combining the individual fitness values together to form a fitness estimate for the whole chromosome. This combination of fitness values, when used to calculate the totalfitness, must preserve the 'spatial' relationships between the individual fitness values.

The genetic algorithm evolves over time and this type of algorithm can have advantages when a sequence of images is presented as the input environment for the genetic algorithm. Unsupervised learning may then be possible, where any descriptions of the shape contour, which are evolving, can follow and, perhaps, identify what sort of motion is present in the image. A change in the current description of a contour can then be associated with various manoeuvres of the object in the image, as seen by the changing description of the object's boundary contour, and thus provide predictions about the future position of the object in the image.

1.6 Aims of the Research Work

The research work described in this thesis had the following aims:

1. *To design a chromosome that was not in the form of a set of parameters for matching masks, filters and model shapes.* Chapters 2, 3 and 4 describe the research work to investigate the form of a chromosome that would contain more information about a shape contour than a set of parameter values.
2. *To investigate the behaviour of a standard genetic algorithm using such a chromosome and the ability of that form of the genetic algorithm to evolve a number of chromosomes that could be used in a subsequent recognition process.* Chapters 5 and 6 describe the research work to investigate the performance of the standard genetic algorithm and the recognition capabilities of the evolved chromosomes.
3. *To analyse the fitness calculation in order to identify a suitable form for the fitness function that would produce high fitness values when two shape contours 'looked alike' to the human observer.* The difficulties associated with the process of matching two shapes have been extensively discussed in the literature (see also Chapter 4). Chapter 7 considers the various characteristics of a variety of fitness functions.
4. *To combine theory (schema theorem) with experimental evidence to show that the standard*

genetic algorithm would, in fact, behave as predicted by the theory and to show that the mutation and crossover probabilities could be adjusted to evolve consistently a number of chromosomes that were able to recognise shape contours in a digitised image. Chapter 8 presents the results of the research work performed to show that the behaviour of the standard genetic algorithm, using the designed chromosomes, did follow the theory and suggests an adaptation of the chromosome to extend the recognition capability of the process.

1.7 Thesis Structure

The research work described in this thesis has investigated the application of genetic algorithm principles and evolutionary programming techniques to the problem of shape contour recognition. The research examined the use of the genetic algorithm principles in order to find out if some of the problems normally associated with shape recognition could be reduced or overcome. Genetic and/or evolutionary algorithms provide a global search mechanism, which can have advantages in finding possible groupings of a contour in such a way that the layout of the chromosomes is a description of the shape when decoded. For recognition, the decoding information will come eventually from a library, which has been built up by the genetic algorithm as it searches through the contour PIP descriptions of the training examples.

Genetic algorithm training can be accomplished by presenting a variety of examples that have sufficient variation to cover the cases under investigation. The encoding of the contour characteristics in a form suitable for input to the genetic algorithm is examined for a variety of evolutionary processes. A new arrival could be inserted into the training set and its contour description evolves in relation to the other examples in the training set. The training set could be considered as the memory for the chromosomes, and contour descriptions would be stored away as the chromosomes for each type of contour evolves. The contour description or feature database could also contain other items of information, e.g. selected moment descriptors, Fourier descriptors,

and number of PIPs on a contour.

Chapter 2 discusses the research performed to investigate the use of a simple genetic algorithm, with chromosomes that were able to control their own motion to detect bright and edge pixels of simple geometric shapes in an image. This research was an adaptation of the evolutionary processes discussed in Dewdney (1989) to the analysis of images. Chromosomes were developed which, in some cases, indicated that edges were present in the image. The research experiments showed that the development of the chromosomes appeared to be little affected by the movement of the objects in the image. As the chromosomes were developing they were able to adjust themselves to the motion of the object(s) in the image. The ability of this configuration to track an object was analysed and its potential for cueing regions of interest and extension to fine grain optical flow were also discussed.

Chapter 3 describes the research performed to investigate the use of genetic programming techniques for evolving chromosomes that are able to follow a closed or open curve. Chromosomes were developed which easily direct a cell to follow a curve, but the characteristics of these chromosomes did not, in general, tell us anything about the curve, i.e. the chromosomes did not 'code' the features of the curve in any way. The inverse transform was also investigated, in which the reproduction of a curve was attempted using only pre-specified moment descriptors, Fourier descriptors, and other characteristics of an object's shape, as 'codes' for the shape of the curve. The chromosomes used in this research had to 'decode' these specified characteristics and must be constructed in a suitable way in order to solve the equivalent of the Travelling Salesperson Problem (TSP), i.e. correctly order the direction information of a curve such that the curve had the specified characteristics.

This problem was equivalent to a 100 city tour and some success was achieved for simple geometric shapes, but the solution was difficult to achieve for more complicated shapes. Another problem was

highlighted by this part of the research work, namely, that it was difficult to use such features as moment descriptors and Fourier descriptors to provide a 'linear' fitness measure for the curve. A linear fitness measure is defined as one that increases (decreases) the fitness value when an intermediate shape is more like (less like) the test shape. An apparent improvement in fitness did not necessarily produce a better shape as seen by the human observer. The TSP was also difficult to solve with any consistency. The equations for calculating normalised central moments are presented for completeness. The difficulties associated with fitness calculations are further discussed in the research described in Chapter 7.

Chapter 4 describes the research that investigated the development of a novel technique for obtaining information about a shape boundary contour that is suitable as input to a genetic algorithm. A normalising stage is discussed which prepared the contour for the detection of PIPs along the contour. The normalising stage was based on Fourier spatial frequency analysis and provided position, rotation, scale and contour starting point normalisation. Closed contours are required for the correct application of the Fourier analysis and the number of spatial frequencies could be varied. The features or descriptions of the contour were calculated in the spatial domain, in order to take full advantage of the relative phase relationships between the various spatial frequencies. The research showed that an improved fitness value could be calculated in the spatial domain, which did not suffer from the relative phase problems associated with the spatial frequencies in the spatial frequency domain. Fourier analysis equations are provided for completeness and for understanding the use that is made of the normalisation process.

The new technique for producing PIP information for a shape contour is described. PIPs on a contour are potentially able to provide the important information needed to define descriptions of the contours. These descriptions should overcome most of the problems associated with the spatial frequency domain, and provide better measurements so that a more 'linear' fitness function can be

achieved. The combination of low-pass filtering of the spatial frequencies present in a contour (available in the spatial frequency pre-processing) and the measurement of PIPs over a variety of scan lengths, provided a form of multi-scale/resolution capability. The contour can be described at various scales and thus information about the contour was available for comparison at these different scales, if required, to sort out any anomalies that may be present in the shape of the contour. Problems associated with classification methods that do not use descriptions are also discussed.

Closed contours were always used in this research so that the periodicity requirement of the Fourier analysis was satisfied. The shape itself need not have closed contours but partial shape contours were analysed as closed contours, to satisfy the conditions required by Fourier analysis.

PIP positions along a contour were used to calculate a description of the contour in terms of the length and direction of the line-segment vector between a pair of PIPs. These line-segment vectors were positioned in a quadrant grid that is centred on the centroid of the original shape contour and bounded by a rectangle whose perimeter contains the entire closed contour and its associated PIP line-segment vector feature information. The quadrants add an extra set of spatial features that improve the definition of the PIP line-segment vectors as input into a genetic algorithm. The start and finish quadrant values were recorded and included in the shape's feature database. One of the problems associated with the genetic algorithm approach concerned the coding which was to be used for the problem's environment, so that the genetic algorithm was applied in an appropriate manner. The research results in Chapter 4 indicated that the above coding did match the genetic algorithm to this particular problem of shape contour recognition. The aspect ratio of this quadrant grid, bounding the coded shape contour and the number of PIPs, can also be included in the description of the contour. The Fourier spatial frequency normalisation process preserved the correct aspect ratio of the original contour.

Chapter 5 discusses the research undertaken to develop an evolutionary method for analysing the PIP descriptions of a contour, and a genetic algorithm was developed which evolved chromosomes that specified various ways to 'observe' how the PIP line-segment vectors were positioned in the quadrant grid. The parameters (crossover and mutation probabilities) associated with the genetic algorithm had to be adjusted such that a number of 'solutions' were found rather than a single 'best' solution. A single 'best' solution may not give sufficient recognition capability to the chromosome, and a number of different ways of observing the contour were usually required to improve the recognition capability of the process. Several experiments are described that investigated a new chromosome 'diversity' measure which enabled suitable values for these genetic algorithm parameters to be found that provided sufficient diversity in the population to evolve the necessary range of chromosomes.

A triangular fitness function was designed and used on a training set of similar shaped contours. This fitness function had the mean and standard deviation for the various contour line-segment vector descriptors as parameters. The same fitness function was used as the basis for shape recognition experiments in Chapter 6. The ability of this fitness function to measure shape fitness was demonstrated in research work reported in Chapter 7.

The genetic algorithm was also able to evolve chromosomes that can identify a mismatch at particular points on the contours in the training set. In this case, the chromosome fitness for reproduction was defined as (1.0 - calculated fitness) and the chromosomes with low calculated fitness (contour mismatch) were then evolved, rather than the chromosomes withhigh calculated fitness (contour match). A distinction has been made between unique (individual) and different solutions. This distinction was used to analyse the performance of the genetic algorithm in relation to the evolution of a number of 'good' solutions.

Chapter 6 describes the research work performed to investigate the ability of the evolved

chromosomes from the genetic algorithm in Chapter 5 and their associated PIP characteristics to provide an effective method for shape recognition that may not necessarily require the full use of conventional classification techniques. The evolved chromosomes define a number of 'good' ways to observe a shape boundary contour in the spatial domain. A new shape boundary contour recognition technique is developed in Chapter 6 that uses the chromosome encoding and the fitness function from the genetic algorithm in Chapter 5. Experimental results are presented to show that the 'best' chromosomes did not necessarily provide the 'best' recognition capability. Thus providing further evidence that a number of 'good' chromosomes are required for reasonable recognition to occur.

The use of the genetic algorithm fitness function for recognition of shape boundary contour 'mismatch' is discussed. This mismatch capability could be used to provide additional information for the recognition process. The research discussed the possibility of 100% recognition, and a recognition matrix was constructed that showed how the chromosomes were used to recognise a set of test shapes. The range of values for the two threshold parameters involved is shown not to be critical for the recognition process. A suggestion is made to investigate further the use of a second concurrent genetic algorithm to adjust these two thresholds so that a possible unsupervised learning capability could be evolved

Chapter 7 considered results from various experimental tests to show the effectiveness of various methods used to calculate the fitness for PIP line-segment vector features of a shape boundary contour. The linearity and symmetry of each method are discussed and this analysis stressed that the fitness function must be investigated and shown to be suitable for the correct operation of the genetic algorithm. A modification to the triangular fitness function, that refined the performance of the genetic algorithm and the recognition process, was proposed as a result of this analysis.

It was noted in this research, that few genetic algorithm references in the literature analysed the

fitness function for correct operation. A fitness function is considered to be suitable for the standard genetic algorithm when it produces a fitness value such that the selection process chooses chromosomes randomly with a probability proportional to their true relative fitness.

Chapter 8 describes various experimental tests that were performed to verify that the Schema Theory did apply to the genetic algorithm configuration from Chapter 5. The schema analysis from the referenced literature is fully presented for completeness. The behaviour of the 'observation' genes in the chromosome in this genetic algorithm is shown to be similar to the behaviour of the schema in a 'standard' or 'canonical' genetic algorithm. The research provides an analysis that indicates the effect that various values for the crossover and mutation probabilities had on the behaviour of the genetic algorithm. An explanation is given for the value of the mutation probability that enabled the genetic algorithm to evolve a number of less fit solutions rather than a single best solution. The epistasis of the chromosomes (effect of one gene on the expression of another gene) is investigated, with reference to the genetic algorithm literature, as a measure of the suitability of the standard genetic algorithm for solving this type of recognition problem. The convergence properties of the standard or canonical genetic algorithm are also discussed and suggestions are made for further research.

Chapter 9 summarises the main achievements of the research work that has been performed and a variety of conclusions are drawn. A number of suggestions for further research are discussed.

A list of referenced papers is provided at the back of the thesis.

2. Bright and Edge Pixel Detection Using Chromosome Position

2.1 Introduction

The first experiments, which were performed to see if a chromosome could contain the necessary information to describe a shape in an image, were considered after reading Dewdney (1989). The article describes the behaviour of cells whose motion is controlled by the bit patterns in the chromosomes for each cell. The motion of the cells in the article congregated around the food sources for the particular cells. The experiment described below, controls the motion of the cells in the same way, but substitutes the grey level and edge values in an image for the food of the cells in the article. Dewdney (1989) discussed simulated evolution "wherein bugs learn to hunt bacteria". A simulation of evolution by M. Palmiter is described in which cells live within a rectangle on which bacteria are continually being deposited. The cells pursue a life dominated by moving and feeding. Each bacteria provides the cell with forty units of energy. This amount of energy is enough to make forty moves. The energy acquired is limited to a threshold and does not benefit the cell until movement reduces the energy of the cell below the threshold again.

Some cells appear to move to the areas of relative abundance quicker than other cells. "It all depends on the moves a cell makes, its search pattern so to speak". Genes that control the way a cell moves are specified and are shown in Figure 2-1.

Direction & (Probability)	Forward (P0)	Right (P1)	Hard Right (P2)	Reverse (P3)	Hard Left (P4)	Left (P5)
X-movement (pixels)	0	2	2	0	-2	-2
Y-movement (pixels)	2	1	-1	-2	-1	1

Figure 2-1: Cell Directions of Motion

"All directions are expressed from the cell's point of view. Normal turns amount to sixty degrees in one or other direction, whereas hard turns are one hundred and twenty degrees". The choice of gene to use is random according to a value of probability stored in the gene itself. After a cell has made

eight hundred moves it becomes 'mature' and is ready to reproduce. Candidates with sufficient energy are selected for reproduction and split into two cells, each of which is given half the energy of its parent. Mutation is applied to each new cell such that one of the genes in each offspring is increased or decreased slightly. Cells start their life jittering, but quickly bobble, tumble, glide and finally turn into 'cruisers'. "Cruisers move forward most of the time but turn every now and then. They almost always move towards denser populations of bacteria. Once this behaviour is established in just a few individual cells, it comes to dominate the entire population because the cruisers end up gathering the lion's share of the bacteria."

This paper considered that the cruisers constitute a species of sorts, "there nonetheless is still some variation within the cruiser population. For example, some cruisers turn more often to the right than to the left, whereas others favour left turns. There are also occasional setbacks. Some cruisers spawn maladapted descendants." A common genetically transmitted disease makes the cell take too many turns in one direction and the cell then becomes a 'twirler'. These cells usually die without having reached the stage where they have enough energy for the selection process.

Dewdney (1989) also described an extension to the simulation that varies the environment to see if more than one species will evolve. The simulation can be run in a mode "in which the screen looks much the same except for a particularly rich patch of bacteria. The bacteria in this patch are replenished at a much higher rate than normal." Cruisers evolve as before but twirlers now make an appearance and "what normally is a disastrous genetic defect is actually now an advantage. Indeed, in the coarse of time those cells with a strong tendency to turn in one direction predominate when positioned in the rich patch of bacteria." A cell that turns frequently in the same direction will now tend to remain in the rich patch of bacteria. It is also noted that the rich patch of bacteria is populated by highly specialised twirlers that follow a specific orbit for many cycles and then "suddenly move just one square away and repeat the orbit, sweeping up bacteria with each shift."

This type of simulation suggested a behaviour that could be used to find an object in an image. The simulation of a cell that is looking for food (energy) could be similar to a process that is attempting to look for 'bright' pixels, i.e. the most energetic pixels in an image. It was hoped that the distribution of bit patterns in the chromosomes would be indicative of the behaviour of the individual cells in such a way that the properties of the shapes in the image might be described in some way.

2.2 Chromosome Encoding of Cell Behaviour

The research used a standard genetic algorithm to study the similarity between food energy and bright image pixels. A fitness-sharing scheme was used to prevent all the cells positioning themselves on the same place in the image.

A cell at any particular position examines the grey level distribution using a simple edge operator and measures the strength of the image gradient at its position. The fitness of a cell is measured by the strength of the edge modified by a crowding factor, which is calculated as a function of the number of nearby cells. The cells are selected from the population in proportion to their corrected fitness. The selected cells reproduced using the crossover and mutation operators. The chromosome of the cell encodes the probability of movement in six directions defined inFigure 2-1. The next generation of cells then move according to the probabilities as specified in the individual chromosomes and position themselves on the image accordingly.

The cycle is repeated as the fittest cells reproduce. The cells, which are initially on the image edges, will generate new cells nearby and after a number of generations the cells position themselves evenly on the image edges. The sharing produced by the crowding mechanism keeps the cells over the edges and avoids premature clustering at single strong edges.

The use of a different operator also allows the cells to position themselves on the image hotspots.

Again, the sharing produced by the crowding method ensures that the cells are distributed over the hotspots. Fairly well defined hotspots may be required to distribute the cells evenly over the image.

The characteristics of the chromosomes were studied to see if a pattern in the values of the chromosome elements could be used to identify any characteristics of the edges or hotspots on the image. Strong values for the forward and backward motion of the fittest cells corresponded to strong edges and hotspots. Different shapes did not show any other strong characteristics in the chromosome values.

A by-product of this particular encoding may be a simple method for cueing regions of interest in an image where the image shows strong edge and hotspot characteristics. As the various objects move their positions within the image, the cells have the ability to follow the strong edges and hotspots. The cells will act as a simple tracker.

The calculation of the fitness of a cell, and in particular the calculation of the crowding factor, uses the most processor resources. Cell reproduction uses the standard genetic algorithm procedures. Elitism improves the speed of convergence of the genetic cycle.

The coding in the chromosome, although suitable for small repositioning of the cell, is not as suitable for following a contour. In general, too many cells will be needed to settle over the whole contour, and the main output from the population would still only be the co-ordinates of the individual cells. A more suitable method is required, which will actually follow the contour but encode its co-ordinates in a more efficient way.

2.3 Description of Genetic Algorithm

The chromosomes are fixed length bit strings, which code the probability of choice for the direction of motion of the cell. The directions are encoded as the amount of rotation the cell performs during

each generation. Reproduction is by population substitution, using only the crossover operator. Two hundred cells evolve for approximately one hundred generations. The cells migrate to the edge and/or bright pixels in less than a generation. The final positions of the cells and their corresponding bit string values are recorded. A fitness-sharing function is used to distribute the cells over the objects in the image in proportion to a particular object's grey level intensity. In this way, convergence and exploitation are avoided and, hence, the maximum amount of exploration of the image is achieved.

The genetic algorithm used for these experiments was an adaptation of the algorithm described by Goldberg (1989) and is described in pseudo-code in Figure 2-2.

For an objective function at cell position *(x , y)*, two fitness functions, f_1 and f_2, are used and are defined by:

$$f_1(x,y) = \sum_{i,j \in S} p(i,j) \tag{2.1}$$

Where *S* is the set of co-ordinates in a 3 by 3 neighbourhood centred at cell position *(x , y)* and *p(i , j)* is the grey level for the pixel at position *(i , j)* and:

$$f_2(x,y) = \sum_{i=1}^{k} \left| p(x-i,y) - p(x+i,y) \right| \tag{2.2}$$

Where k = 1, 2 or 3.

Equation (2.1) is an elementary brightness detector and equation (2.2) is an elementary vertical edge detector.

```
Genetic Algorithm Pseudo Code

1. Initialise the population of N cells at random positions (x,y) in
   the image.
2. Select two cells with probability proportional to their fitness
   values (Biased roulette wheel selection).
3. Select a crossover point, in the chromosome bit strings of the
   cells, with uniform probability.
3. Apply a crossover operator to the parent cells and produce two
   cells with new bit string coding.
4. Decode the new chromosome bit strings and calculate the
   probability for the directions of motion of the new cells.
5. Select the direction of motion for the new cells by 'biased
   roulette wheel selection'.
7. Move the cells to their new positions.
8. Calculate the fitness for the new cells and share fitness
   with the neighbouring cells.
9. Calculate and record the cell statistics.
10. Repeat steps 2 to 9 N/2 times in order to replace
    completely the fixed size population for each generation.
11. Repeat steps 2 to 10 for n generations.
```

Figure 2-2: Genetic Algorithm Pseudo Code

In order to distribute the cells in proportion to the grey level intensity of the objects in the image, and avoid convergence onto a few image space regions, some form of distribution of fitness is required. Goldberg and Richardson (1987) defined a sharing function *sh(d)* as a function of the variable *d* with the following three properties:

$$0 \le sh(d) \le 1, \textit{for all } d \tag{2.3}$$

$$sh(0) = 1 \tag{2.4}$$

$$\lim_{d\to\infty} sh(d) = 0 \tag{2.5}$$

A power law function is chosen by them such that:

$$\begin{aligned} sh(d) &= 1 - \left(\frac{d}{\sigma_{share}}\right)^{\alpha}, \; d < \sigma_{share} \\ &= 0 \;, \; otherwise \end{aligned} \tag{2.6}$$

Where σ_{share} and α are constant values for a particular experiment.

The shared fitness f_p' of the p^{th} cell is defined as its potential fitness f_p (calculated from f_1 and f_2) divided by its niche count m_p'

$$f_p' = \frac{f_p}{m_p'} \tag{2.7}$$

The niche count m_p' is given by:

$$m_p' = \sum_{q=1}^{N} sh(d_{pq}) \tag{2.8}$$

Where N is the number of cells in the population and d_{pq} is the Euclidean distance between the p^{th} and q^{th} cells.

The directions of motion of a cell are chosen according to a probability, which is coded into the fixed length chromosome bit string associated with each cell. The sum of the probabilities is set equal to 1.0. The directions associated with each probability are adopted from Dewdney (1989) and are shown in Figure 2-1.

These directions define the amount of translation and rotation performed by the cells during each generation. P1 and P2 provide clockwise (Right and Hard Right), and P4 and P5 an anti-clockwise (Hard Left and Left) rotation of the cell. P0 moves the cell in the forward direction, while P3 reverses the direction of motion of the cell.

2.4 Experiments and Results

Various experiments were performed with simple geometric shapes, using a grey scale from 0 to 255, in a square image of 100 by 100 pixels. Two hundred cells were developed for a succession of one hundred generations. The crossover probability was set at 0.8, with σ_{share} having a value of 15.0 pixels and α equal to 1.0. The following simple tests were repeated for the fitness functions f_1 and f_2 as defined above:

1. One square object (A) centred at pixel co-ordinates (50, 50), with dimensions of 41 pixels by 41 pixels having a grey level of [127] and zero background.
2. One square object (A), with centre and dimensions as above, having a grey level of [200] and a grey level of [63] as a uniform background.
3. Two square objects (B & C), with dimensions of 21 by 21 pixels, having grey levels [127] and [255] respectively and zero background. The centres of the objects are the pixel co-ordinates (25,25) and (75,75) respectively.
4. A repeat of test 3 with the lower intensity object (left) moving down the image and the higher intensity object (right) moving up the image. Both objects moved 1 pixel every 5 generations.
5. A repeat of test 2 with added uniformly distributed noise, having a standard deviation of±11.6 grey levels.

The positions of the objects in each image and the final distribution of the cells for each of the above tests are shown in Figure 2-3.

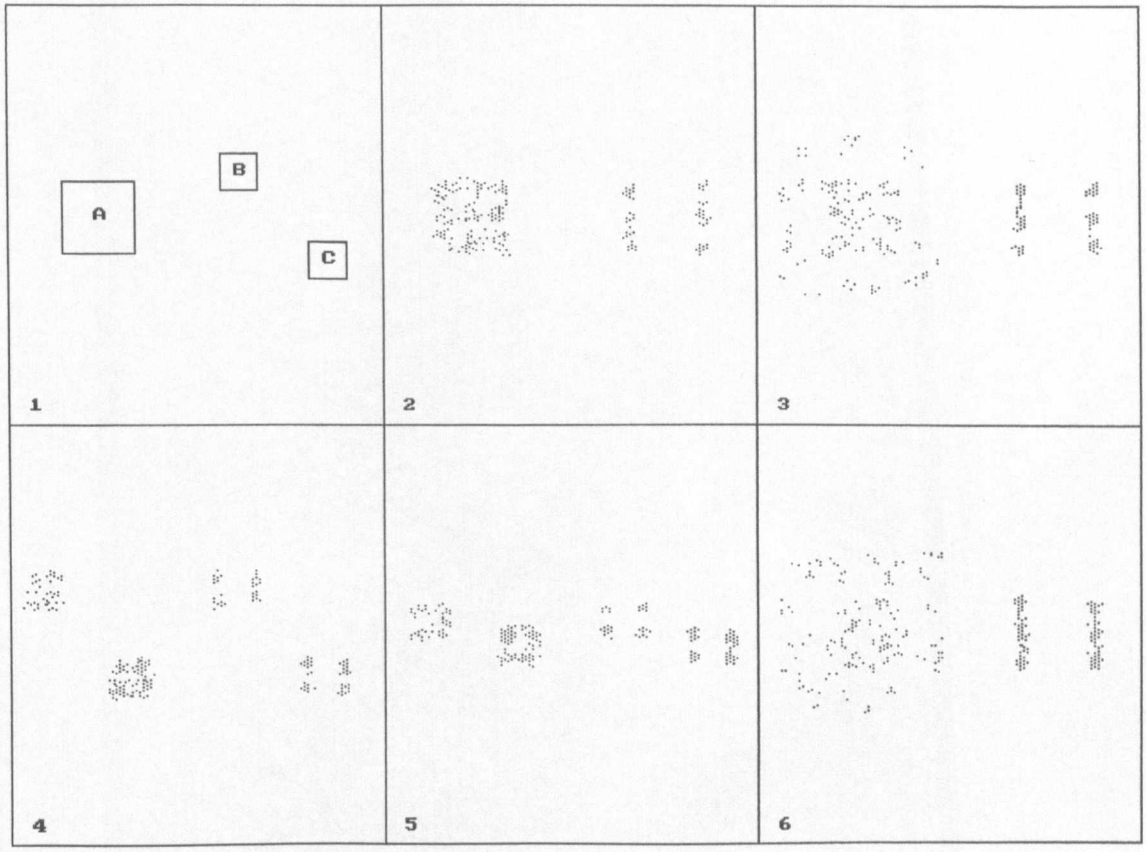

Figure 2-3: Final Distribution of Cells

For the single object tests, with the fitness function f_1, the cell distribution was correctly centred on the object. The mean (x , y) co-ordinates for the cells were within 2 pixels of the centre of the object. The cells positioned themselves over the total area of the object in numbers proportional to the relative brightness of the object and the corresponding background. The standard deviation of the cell (x , y) co-ordinates was proportional to the size of the objects. For the two object tests, with fitness function f_1, the cells correctly positioned themselves over the whole area of each object. The number of cells in each object was in proportion to their relative brightness, i.e. the population fitness was shared between the objects and/or backgrounds (Figure 2-4).

For the single object cases, using the fitness function f_2, the cells correctly positioned themselves onto the edges of the objects in approximately equal numbers. For the two object tests, with this fitness function, the cells correctly moved to the edges of the objects in numbers approximately proportional to the relative grey levels of each object (Figure 2-5).

Test	Object	Mean Co-ordinates		Standard Deviation		Number on Each Object
		X	Y	X	Y	
1	A	51.0	50.0	13.2	13.1	200
2	A	48.0	48.0	14.0	13.6	130
3	B	27.0	26.0	5.9	7.6	59
	C	75.0	75.0	6.9	6.6	141
4	B	25.0	43.0	7.3	6.5	60
	C	75.0	56.0	7.0	6.9	140
5	A	49.0	51.0	13.4	13.1	129

Figure 2-4: Final Distribution of the Cells in Each Image for F_1

Test	Object	Mean Co-ordinates		Standard Deviation		Number on Each Edge
		X	Y	X	Y	
1	A(L)	30.0	51.0	1.5	12.5	88
	A(R)	71.0	52.0	1.4	15.1	112
2	A(L)	30.0	50.0	1.5	14.1	96
	A(R)	71.0	49.0	1.6	12.8	104
3	B(L)	14.0	23.0	1.8	7.2	28
	B(R)	35.0	24.0	1.3	5.5	27
3	C(L)	64.0	74.0	1.6	7.2	68
	C(R)	86.0	75.0	1.6	6.9	76
4	B(L)	15.0	45.0	1.8	6.1	31
	B(R)	36.0	44.0	1.7	7.4	35
4	C(L)	64.0	55.0	1.5	6.8	60
	C(R)	86.0	59.0	1.6	6.6	74
5	A(L)	29.0	49.0	1.7	13.8	95
	A(R)	71.0	50.0	1.6	13.3	105

Figure 2-5: Final Distribution of the Cells in Each Image for F_2

The direction of motion probabilities as stored in the chromosome of each cell, which were developed for the fitness function f_1, showed a general rotation in the clockwise and anti-clockwise direction. A bias towards P3, which indicates a reversal in the direction of motion of the cell, was also apparent. Development of a large value for P3 is a characteristic of an edge in the image (Figure 2-6). The fitness function f_2 developed cells with a strong bias to reverse their direction of motion (P3). Clockwise (P2) and anti-clockwise (P4) rotation were also developed depending on the conditions of the test (Figure 2-7). The development of the direction of motion probabilities and the number of cells on each object were similar for objects of different grey level intensity and were influenced by the fitness function rather than the size of the object.

Test	Object	P0	P1	P2	P3	P4	P5
1	A	0.11	0.2	0.07	0.24	0.22	0.16
2	A	0.17	0.23	0.10	0.10	0.24	0.16
3	B	0.14	0.10	0.10	0.26	0.21	0.19
	C	0.13	0.11	0.10	0.24	0.24	0.18
4	B	0.02	0.19	0.07	0.30	0.14	0.28
	C	0.02	0.18	0.10	0.29	0.14	0.27
5	A	0.09	0.28	0.10	0.20	0.18	0.15

Figure 2-6: Final Distribution of the Cell Direction of Motion Probabilities for F_1

Test	Object	P0	P1	P2	P3	P4	P5
1	A(L)	0.04	0.02	0.12	0.72	0.03	0.07
	A(R)	0.02	0.03	0.12	0.72	0.04	0.07
2	A(L)	0.04	0.05	0.13	0.4	0.35	0.03
	A(R)	0.01	0.05	0.14	0.41	0.36	0.03
3	B(L)	0.06	0.05	0.38	0.42	0.06	0.03
	B(R)	0.04	0.06	0.39	0.40	0.09	0.02
3	C(L)	0.03	0.04	0.39	0.42	0.10	0.02
	C(R)	0.04	0.04	0.39	0.42	0.09	0.02
4	B(L)	0.03	0.08	0.17	0.39	0.27	0.06
	B(R)	0.06	0.08	0.16	0.36	0.27	0.07
4	C(L)	0.03	0.08	0.17	0.37	0.29	0.06
	C(R)	0.03	0.05	0.17	0.38	0.30	0.07
5	A(L)	0.08	0.03	0.16	0.39	0.27	0.07
	A(R)	0.09	0.04	0.15	0.39	0.27	0.06

Figure 2-7: Final Distribution of Cell Direction of Motion Probabilities for F_2

The cells successfully tracked the translational motion of the objects, while maintaining the correct proportion of cells on each object, as in the stationary case. The root mean square (*rms*) tracking errors, in the y direction, were measured as 1.31 pixels for the left object and 0.90 pixels for the object on the right. Adding noise to the simple geometric shapes in an image did not disturb the positions or distribution of the cells significantly. The cell ratios for the two object case, test 3 and 4, displayed an oscillatory behaviour, caused by the 'division term' in the sharing function (Equation (2.7)). The mean and standard deviation of the cell ratio for the two stationary objects were 2.08 and 0.38 respectively and 2.08 and 0.36 for the moving objects case. The mean value for the cell ratio corresponded to the ratio (2.0) of the grey levels of the objects.

This experiment also shows that the chromosomes can evolve as the object(s) change their positions in the image, i.e. in this case, the development of the chromosome is not significantly affected by

the motion of the objects. The development of the chromosomes in these simple tests was not responsive to different characteristics of the fitness functions. Large values of P3 and P0 did indicate that edges were present in the image, but the chromosomes did not develop any particular pattern that would give more information about the characteristics of the objects in the image.

2.5 Summary and Conclusions

The set of experiments above were presented in a paper at an IEE colloquium (Egan and Picton, 1994) and could possibly be extended into simulating fine grain optical flow with the appropriate fitness functions. This research suggested that controlling the motion of the cells, using the above chromosomes, can be useful for cueing areas of interest in an image, and especially for a sequence of images, where the objects might change their position, orientation or scale. Skew effects could also be taken care of by a suitable fitness function. Unfortunately this form of coding the chromosome only seems to indicate the direction of motion of the fittest cells but does not in itself provide any form of description of an objects shape. Other edge operators could be tried in the fitness function, such as a second differential, which would enable the cells to track the zero-crossing contours in the image. This form of chromosome coding acts as a good tracker, that could be implemented in a fine grained algorithm, but the chromosomes only describe the motion of the cells and not the characteristics of the object being tracked. The sharing process, in practice, would also be expensive on processor timings.

The technique developed in the research work described in this chapter could possibly be extended to an application involving optical flow. Schiff (1994) discussed a hardware implementation of a biological neural system for target localisation. The hardware implementation is derived from a mathematical model of a biological system. "The visual systems of animals have evolved through many millions of years and thus have reached a high degree of optimisation for each specific function. Nervous processing is massively parallel. In vision, many photo-transducers send

electrical inputs simultaneously into several nervous sub-systems. Each sub-system serves a different final purpose and receives differently pre-ordered sets of inputs." This paper studies one of these sub-systems that analyses the position, size and motion of a target quickly and accurately. According to the model used, incomplete information can be completed because of the high redundancy and simultaneous inputs to many photoreceptors.

Huang (1995) discussed optic flow field segmentation and motion estimation using a genetic algorithm, and points out that optic flow motion analysis basically has two related stages. The first stage involves the computation of the optic flow field and the second stage requires a meaningful interpretation of the calculated optic flow field regarding the underlying object structures and motion parameters.

The completion of the above sequence of experiments, using the position of a chromosome, suggested a further set of experiments, described in Chapter 3. These tests were designed to investigate if training a cell to follow a contour would provide chromosomes that could contain information about the characteristics of the traced contour.

3. Contour Tracing Using Chromosome Motion

3.1 Introduction

Training a cell to follow a contour, by learning specific sequences of movements, could possibly result in information about the contour being stored in the chromosome of the cell. The chromosome of the cell could be in a form suitable for the application of the methods described by J. Koza for genetic programming (see Koza 1992 and 1994). The chromosome can be structured as a set of functions with parameters. The simple set chosen for the experiments, which are discussed in this chapter, are such actions as turn right, left and move forward. The chromosome represents a hierarchy of actions, which are repeated continuously to traverse the contour.

Another use for this possible coding of the chromosome is in the form similar to that required by a Travelling Salesperson Problem (TSP). After a cell has successfully traced a contour, the contour is coded as the set of paths, which can be traversed on the hierarchical tree. Reproducing the curve then becomes an ordering problem or TSP. Gu and Huang (1994) discussed a method for an efficient local search using search space smoothing. "Due to the rugged terrain surface of the search space, an algorithm often gets stuck at a locally optimum configuration." In a parameter space, by altering the shape of the objective function, the original problem instance is transformed into a series of gradually simplified problem instances with smoother terrain surfaces. "... an instance with the simplest terrain structure is solved first, the original problem instance with more complicated terrain structures is solved last, and the solutions of the simplified problem are used to guide the search of the more complicated ones." It is suggested that a significant improvement in performance of conventional local search algorithms can be achieved. The power of the search space smoothing technique relies on "the simplified problem instances and the effectiveness of using the intermediate solutions to guide the search of the increasingly complicated problem instances."

Experimental results presented in this chapter show that chromosomes could be developed which are able to trace a contour and provide a variety of hierarchical trees for doing so. Some movements

of the cells are very efficient others are more cumbersome. The genetic programming method suffers from the problem of a variable length chromosome, with the subsequent artificial limit placed on the maximum length allowed. The solution to the TSP is not very satisfactory and cannot be achieved in most cases. The genetic programming method is ideal for developing a sequence of actions or functions, which can be repeated to follow a curve or a trajectory using the chromosome to hold the sequence of actions or functions. The time taken to find the solution for tracing a set of contours is considered to be excessive. Although the processing could be done in parallel, it is a non-trivial problem to ensure that any particular chromosome stayed on its correct contour.

Reconstructing the contour and comparing with a prototype has possibilities for recognition of a shape but the calculation of a suitable fitness function can be a problem. As the generated contour is constructed from the fitness calculated by moment descriptors (see section3.5 below), Fourier descriptors (see Chapter 4) or pattern matching, the fitness must be linear, i.e. higher fitness must indicate a shape which is, in fact, nearer the solution than the previous version. The various problems associated with the generation of a linear fitness function for two dimensional (2D) contours are discussed in Chapter 7.

3.2 Description of Genetic Programming

Kinnear (1994, page 6) suggests the following as a definition of the genetic programming paradigm. Genetic programming is an offshoot of genetic algorithms, but in some ways is more of a generalisation than a specialisation of its parent discipline. The following are the distinguishing features of genetic programming:

1. *Non-linear and usually tree-structured with genetic material.* While some genetic algorithms have genetic material that is non-linear, linear genetic material remains the rule for genetic algorithms. However, genetic programming almost always operates on non-linear genetic material, and is usually explicitly tree-structured.

2. *Variable length genetic material.* Genetic programming almost always operates on genetic material that can vary in size. For practical reasons, size limitations on its growth are usually implemented, but these usually allow considerable growth from the original randomly generated population.
3. *Executable genetic material.* Genetic programming is the direct evolution of *programs* for computers. Thus, in almost all cases the genetic material that is evolved is in some sense executable. Executable is not a precise term. Usually the structures are interpreted by an interpreter, in a language identical or very similar to an existing computer language, or in a very specialised task-orientated language designed for the problem in hand. However, in almost all cases, there is some concept of *executing* the genetic material by some sort of interpreter in order to perform the desired function from which the fitness is derived.
4. *Syntax preserving crossover.* While many crossover operators for genetic programming have been reported, in most cases they are defined so that the syntactic correctness of the program is preserved.

Koza (1992) demonstrated that "an automatic, domain-independent method can genetically breed computer programs capable of solving, or approximately solving, a wide variety of problems from a wide variety of fields." and defines the following five major steps when applying genetic programming to a problem:

1. *Set of terminals.* The set of terminals, together with the set of functions, are the ingredients from which genetic programming attempts to construct a computer program to solve, or approximately solve, the chosen problem.
2. *Set of primitive functions.* The set of functions must be identified that are to be used to generate the mathematical expressions that attempt to fit the measured finite sample of data. Each function should be able to accept as its arguments any value and data type that may possibly be returned by any other function in the set.

3. *Fitness measurement.* Each individual computer program in the population is measured in terms of how well it performs in the particular problem environment. The nature of the fitness measure varies with the problem. For some problems, it may be appropriate to use a multi-objective fitness measure, incorporating a combination of factors such as correctness, parsimony, or efficiency.
4. *Parameters for controlling the program evolution.* The Darwinian principal of reproduction, survival of the fittest, and the genetic operation of crossover are used to create a new offspring population of individual computer programs from a current population of programs. The mutation operator may also be used in genetic programming.
5. *Method for designating the result and the criterion for terminating the evolution* The best individual appearing in any generation of a run i.e. the best-so-far individual is typically designated as the result produced by the run of genetic programming. Termination usually occurs after a set number of evolutionary generations, or when the fitness of the best-so-far solution fails to increase over a set number of generations.

Koza (1992) also noted the following characteristics of genetic programming:

1. The hierarchical character of the computer programs produced is an important feature of genetic programming. In many cases the results produced are default hierarchies, prioritised hierarchies of tasks, or hierarchies in which a particular behaviour subsumes or suppresses another.
2. The dynamic variability of the computer programs that are developed along the way to a solution is also an important feature of genetic programming. It is often difficult and unnatural to try to specify or restrict the size and shape of the eventual solution in advance.
3. The computer programs produced by genetic programming consist of functions that are natural for the problem domain and there is an absence of, or relatively minor roles for pre-processing inputs and post-processing of outputs.

4. The structures (chromosomes) undergoing adaptation are active. They are not a passive encoding of the solution to the problem. Instead, given a computer on which to run, the structures in genetic programming are active structures that are capable of being executed in their current form.

Genetic programming can be scaled up to more difficult problems by means of the automatic discovery of functional sub-units, known as Automatically Defined Functions (ADFs). Koza (1994) discussed the implementation of ADFs within the context of genetic programming by establishing a constrained structure for the individual programs in the population. "Each program in the population contains one or more function-defining branches and one or more main result-producing branches. The result-producing branch usually has the ability to call one or more of the defined functions. Genetic programming will evolve a population of programs, each consisting of a function-defining branch and a result-producing branch. Whether or not the defined function will be actually called is not predetermined, but instead, will be determined by the evolutionary process. Since a constrained syntactic structure is involved, crossover must be performed so that the syntactic structure of all offspring is preserved."

3.3 Contour Tracing Using Genetic Programming

The initial research work on this aspect of genetic algorithm investigation was based on developing a program, which would evolve the 'Santa Fe Ant' (see Koza, 1992) using genetic programming techniques (Figure 3-1). A limit was imposed on the 'action depth' and the number of steps allowed to be executed. The 'action depth' controls how far along the chromosome the actions of the cell are allowed to proceed. The best 10% from each generation were automatically saved. Chromosomes were selected for reproduction in proportion to their fitness and the whole population was replaced at each generation. Single point crossover was applied without mutation. No limit was imposed on the length of the chromosome. As discussed above, one of the problems associated with genetic

programming is that the length of the chromosome is able to vary. Eventually the chromosome may have to be truncated to avoid program memory overflow. The truncation method has to be chosen carefully to avoid interference with the operation of the crossover operator.

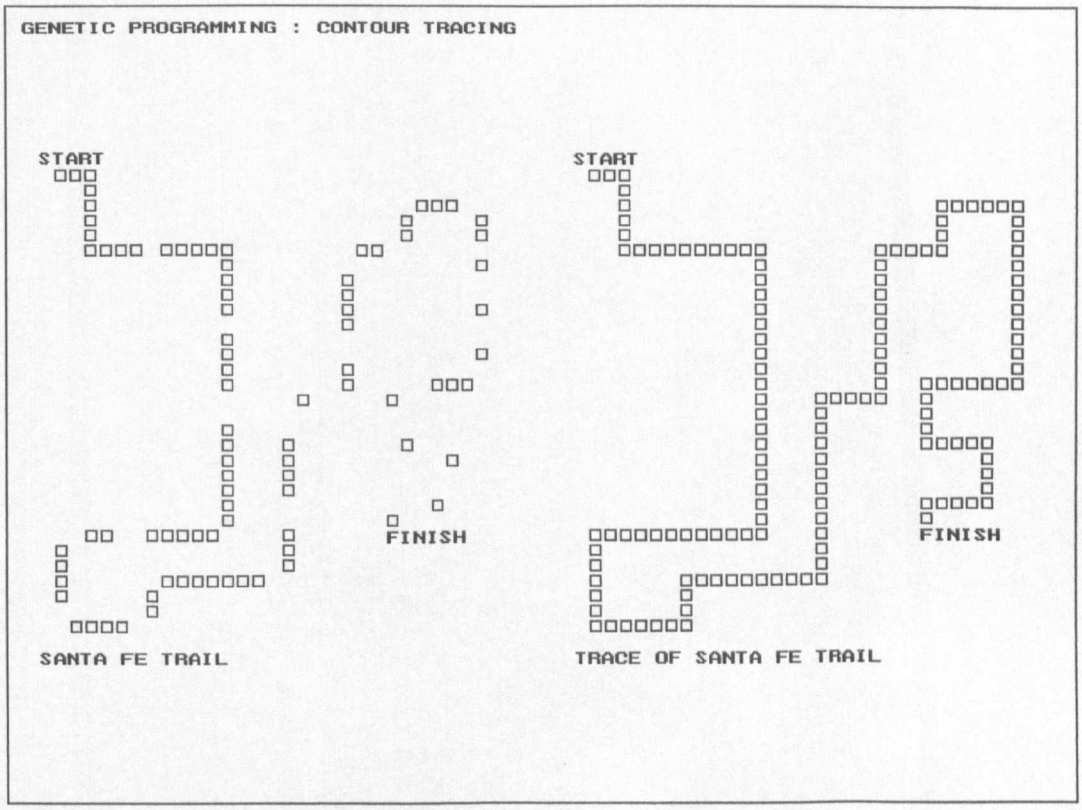

Figure 3-1: Santa Fe Ant Trail: Contour Following

The chromosome represents a hierarchical set of actions with parameters, which are activated a number of times each generation in an attempt to trace the contour successfully. Sensor inputs from the environment enable the hierarchical tree to make decisions. The basic functions used in this part of the study are sensor, right, left, advance and sequence. The sensor function looks ahead and activates action 1 if the pixel is illuminated and action 2 if the pixel is not illuminated. The sequence function performs the parameter functions in the sequence specified by the order of its parameters. The right and left functions turn the direction of the sensor to look ahead in the relevant direction. The advance function moves the action one step in the current direction.

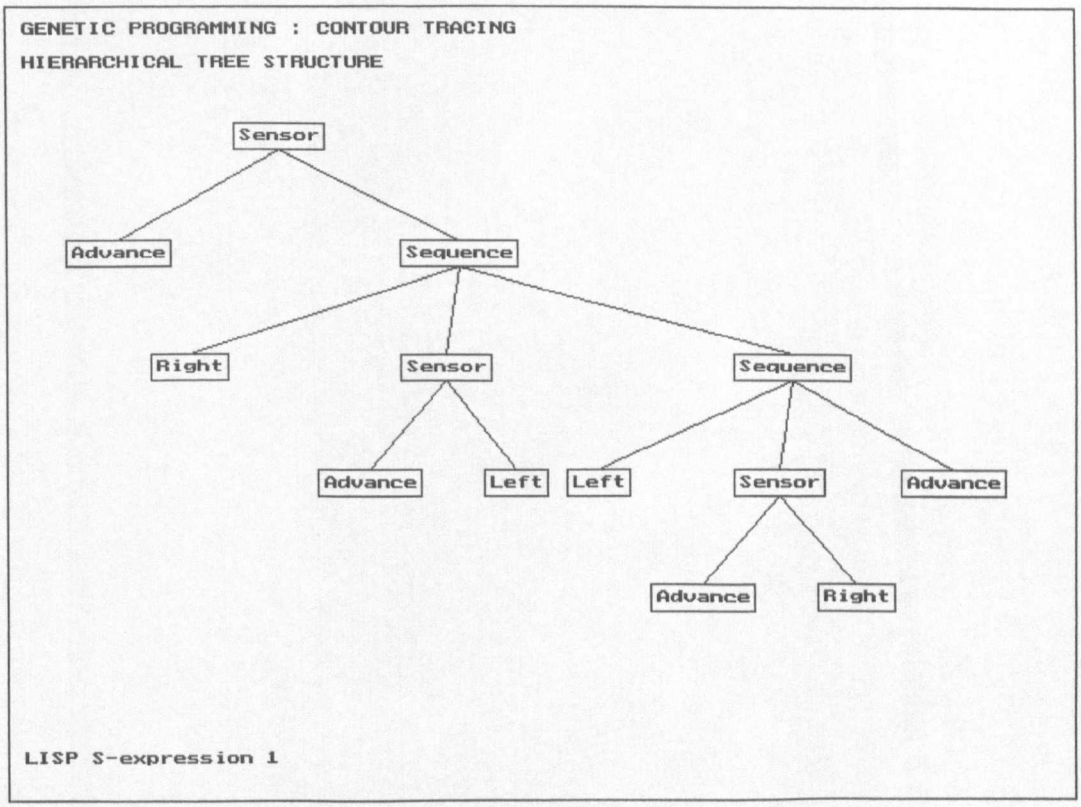

Figure 3-2: Contour Tracing: Action Sequence 1

The 13-action solution followed the trail in fewer steps than J. Koza's solution, and finished on the last path element of the trail. Koza's solution also executed 13 actions, but this solution (Figure 3-2) did not evolve during any run of the program. A program was then developed to identify all the separate paths through the hierarchy defined by the chromosome, in preparation for the contour encoding investigation discussed in section 3.4.

Many solutions were obtained, with two of the best having 11 and 13 actions respectively (see Figure 3-3 and Figure 3-4).

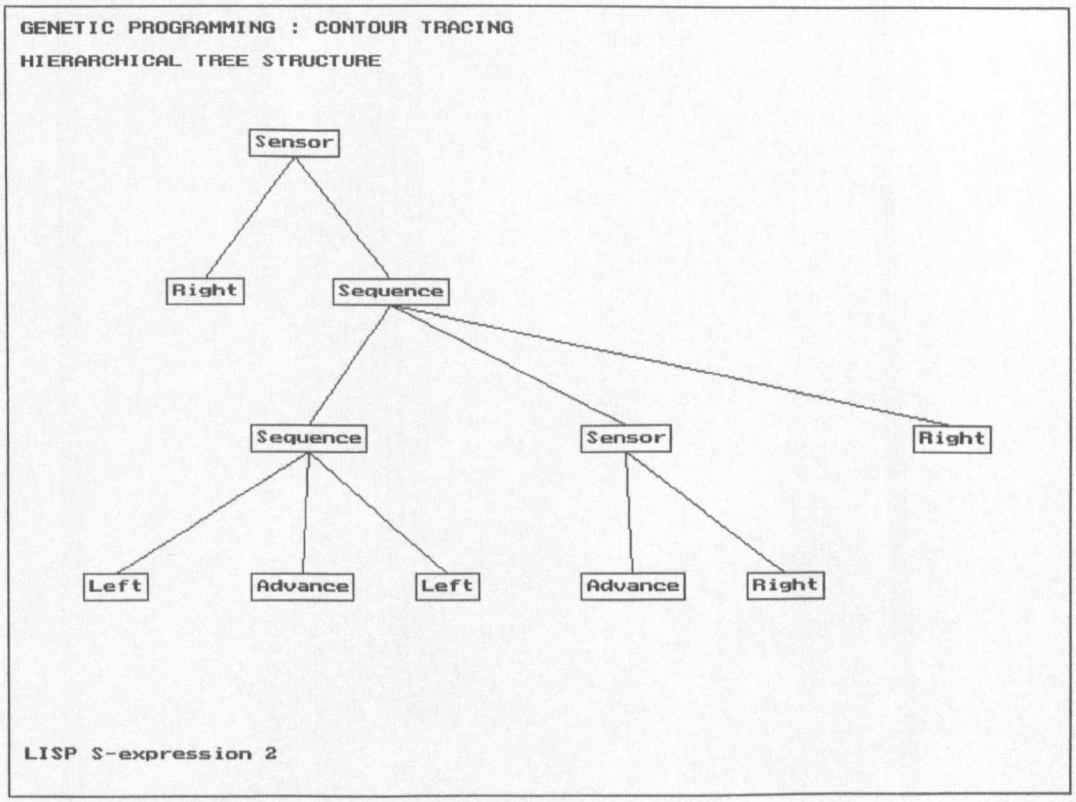

Figure 3-3: Contour Tracing: Action Sequence 2

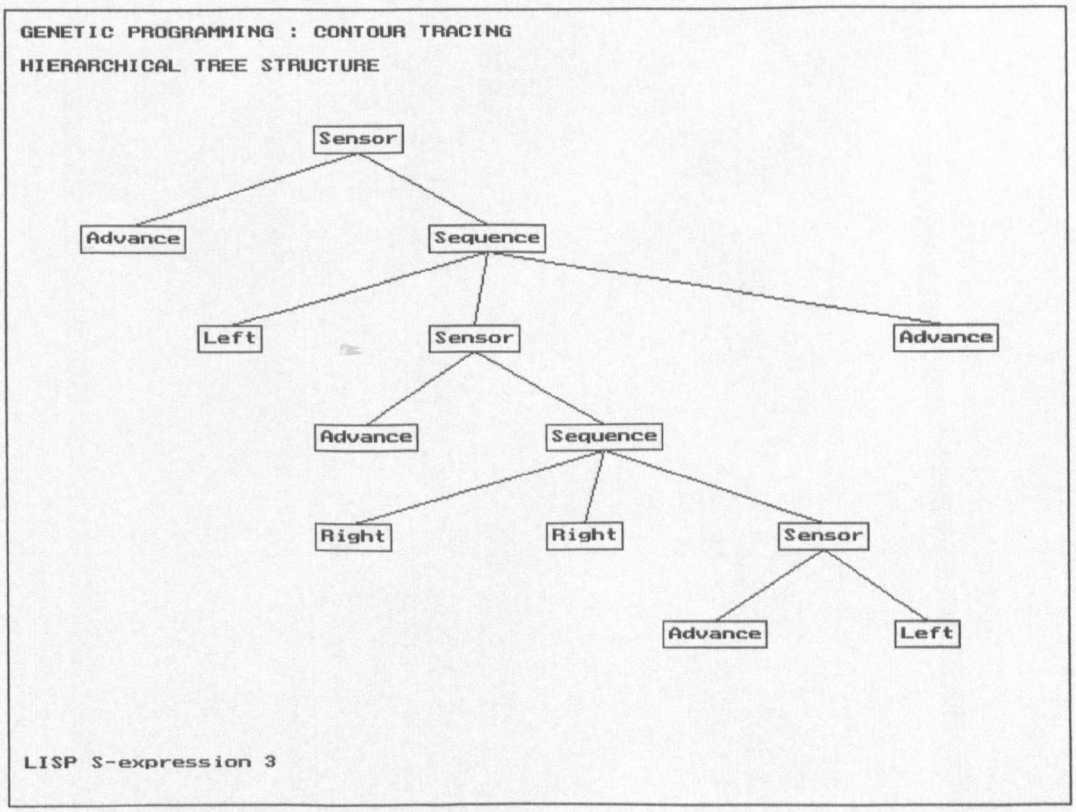

Figure 3-4: Contour Tracing: Action Sequence 3

3.4 Contour Encoding Using Genetic Programming

The genetic algorithm was then applied to the problem of reproducing a curve from a defined hierarchical tree of paths, functions and parameters. The chromosome was encoded as a set of path numbers as produced from the above path-identifying program. The fitness of each cell after performing the actions specified by its chromosome, was calculated as a function of the final centre of gravity of the path that was traced, the number of each action, the number of paths, and a normalised set of moments (Figure 3-5).

```
SUN 10-OCT-1993 :: 10:50:15
          Environment File Name: santafe.pic
          SPD         File Name: ant.spd  (Sequence Path Database)
           0 : Initial x Co-ordinate
           0 : Initial y Co-ordinate
           2 : [ East  ] Initial Direction
          23 : Final x Co-ordinate
          24 : Final y Co-ordinate
           4 : [ South ] Final Direction
           5 : Number of Paths
          53   { Advance                }
           0
          11   { Right Advance Advance }
          10   { Left  Advance Advance }
          50   { Advance                }
           7 : Number of Moments
          -0.1695791
           0.5404679
           0.5501451
          -0.1495313
          -0.0369755
          -0.0156923
          -0.0300447
          15.5793103 : Xbar
          15.6000000 : Ybar
          20.3635374 : Ellipse Major Axis
          14.7608223 : Ellipse Minor Axis
           0.1535516 : Irradiance
```

Figure 3-5: Trail Encoding File

The TSP program used this information to compare these reference parameters with those calculated for each chromosome in the population, and calculated the fitness of each chromosome using an *rms* error method for each of the parameters shown in (Figure 3-5). Normalised Central Moments (NCMs) were chosen at this stage in the study because the test cases were not necessarily

closed curves and it was considered that Fourier descriptors would not be appropriate (see Chapter 4). The crossover operator chosen for these tests was modelled on the references to genetic algorithms that have been used to solve the TSP, because the paths taken and the actions that are performed have to be in a particular order for a correct reproduction of the original curve.

Gonzalez (1987, page 419) gives a definition of the standard moment calculations that are reproduced below for completeness. Given a two-dimensional continuous function $f(x,y)$ the moment of order $(p+q)$ can be defined by the relation:

$$m_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p y^q f(x,y)\,dx\,dy \tag{3.1}$$

for $p,q = 0, 1, 2, \ldots$

If $f(x,y)$ is piecewise continuous and has non-zero values only in a finite part of the x - y plane, then moments of all orders exist and the moment sequence (m_{pq}) is uniquely determined by $f(x,y)$. Conversely (m_{pq}) uniquely determines $f(x,y)$. The *central moments* can be expressed as:

$$\mu_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,dx\,dy \tag{3.2}$$

where:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{3.3}$$

For a digital image, equation (3.2) becomes:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x,y) \tag{3.4}$$

The *normalised central moments*, denoted by:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{3.5}$$

where:

$$\gamma = \frac{p+q}{2} + 1 \tag{3.6}$$

for $p+q = 2, 3, \ldots$

Teague (1980) noted that if only moments up to the second order are considered, the original image is completely equivalent to a constant irradiance ellipse (Figure 3-6) having a definite size, orientation and eccentricity and centred on the image centroid.

The parameters, a, b *and* ϕ, of this 'image ellipse' are given by:

$$a = \left(\frac{\mu_{20} + \mu_{02} + \left[(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \right]^{\frac{1}{2}}}{0.5\mu_{00}} \right)^{\frac{1}{2}} \tag{3.7}$$

$$b = \left(\frac{\mu_{20} + \mu_{02} - \left[(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \right]^{\frac{1}{2}}}{0.5\mu_{00}} \right)^{\frac{1}{2}} \tag{3.8}$$

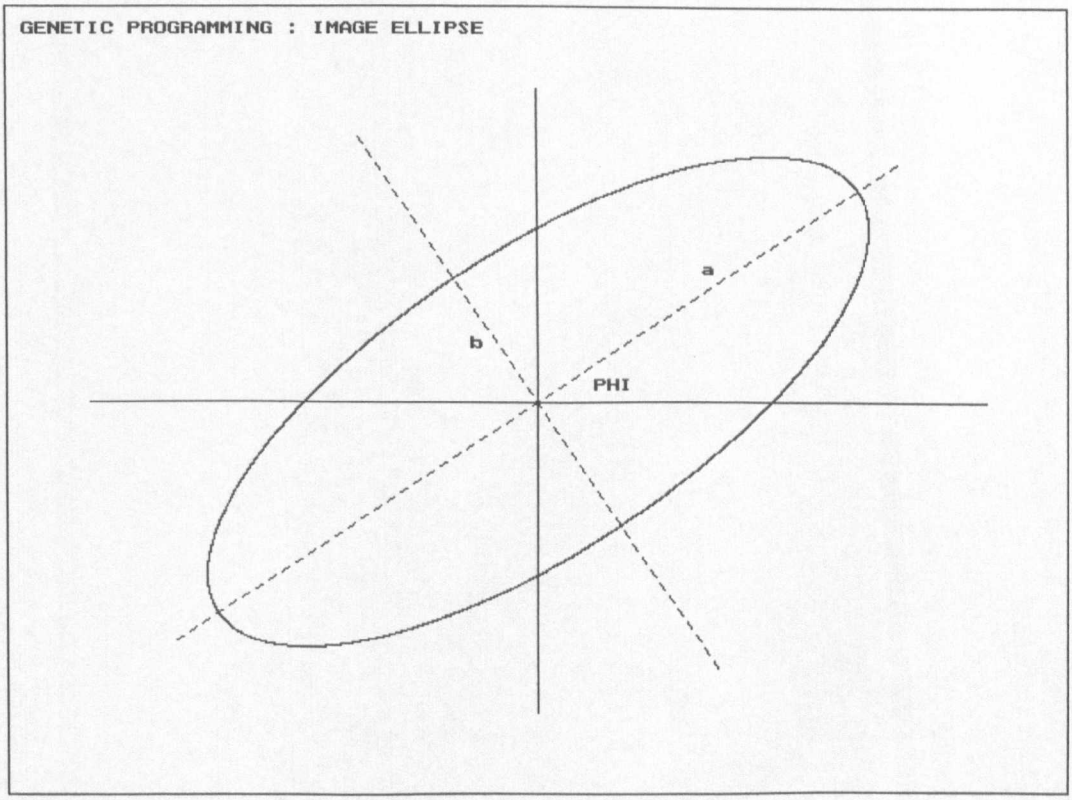

Figure 3-6: Image Ellipse: Moments Calculation

for the semi-major (a) and semi-minor (b) axes respectively.

The angle, ϕ, for the ellipse tilt is given by:

$$\phi = 0.5 \tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{3.9}$$

The irradiance, F, of the image ellipse is defined by:

$$F = \left(\frac{\mu_{00}}{\pi ab}\right) \tag{3.10}$$

inside the ellipse and zero outside the ellipse.

There is an ambiguity in the tilt angle ϕ of the ellipse that is resolved by always choosing ϕ to be

the angle between the x-axis and the semi-major axis, i.e. by definition $a \geq b$ and by choosing the principal value of the function arc tan such that:

$$-\frac{\pi}{2} \leq \tan^{-1}(x) \leq +\frac{\pi}{2} \tag{3.11}$$

This paper also pointed out that the image ellipse exists with real a and b if, and only if, the function $f(x,y)$ is non-negative, which is true for irradiance distributions, but not for amplitude distributions.

Sardana (1994) noted that the value of γ when considering the edge pixels of a silhouette is given by:

$$\gamma = p+q+1 \tag{3.12}$$

which is different than that given in equation (3.6). This difference should be taken into account when using moment descriptors on the boundary pixels of a shape silhouette contour.

3.5 Contour Encoding Results and Discussion

A square trail was reproduced on many occasions but very little success was achieved for a simple cross shape (Figure 3-7).

Two principal problems occurred. Firstly, the fitness calculation is not necessarily linear, in the sense that higher fitness sometimes gives a worse shape as seen by the human observer. Secondly, the crossover operators used, which were obtained from the TSP literature, cannot necessarily make the appropriate rearrangements to the chromosome that are required in the final stages of the development of the shape.

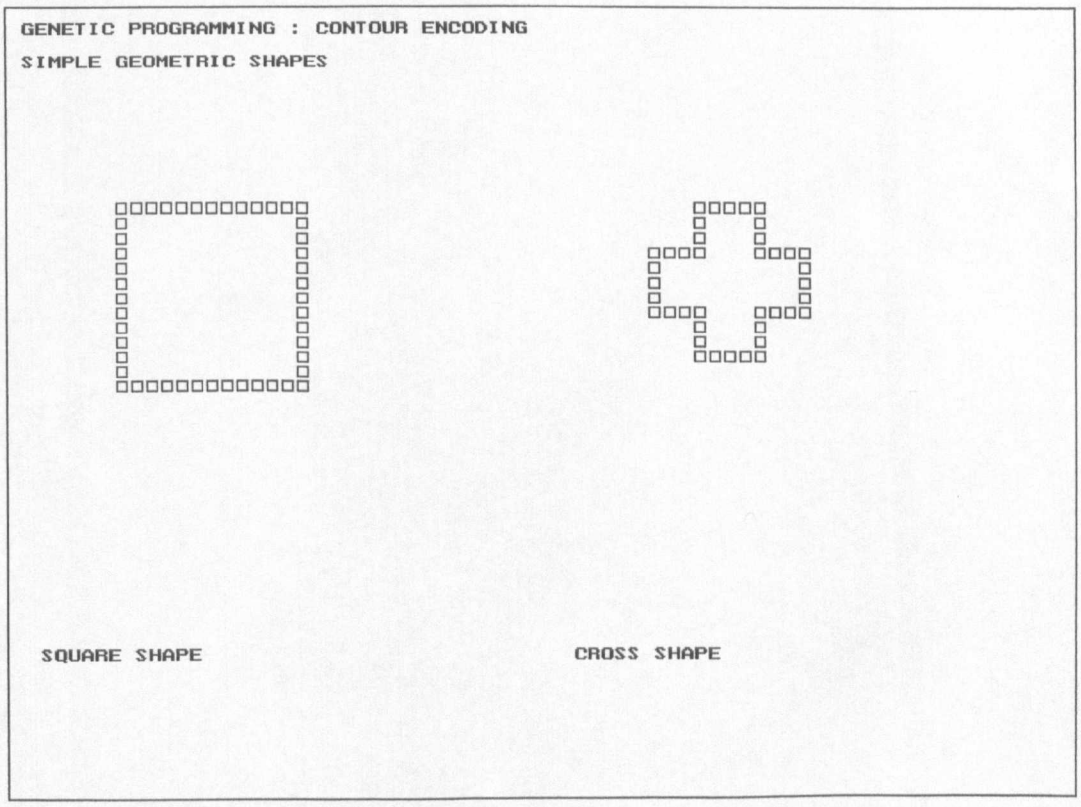

Figure 3-7: Simple Geometric Shapes

Co-occurrence of the various paths, Fourier descriptors and moment descriptors, were also investigated as candidates for inclusion in the fitness function. The TSP ordering always dominated the final stages of the evolution of the solution curve.

In general, the various experimental tests showed that the contour of a shape, such as a square, could easily be reconstructed. Contours of shapes like the cross could be reproduced, but not with any consistency. The Santa Fe Ant Trail was never reproduced by any of the combination of genetic algorithm parameters used. The Santa Fe Ant Trail image is equivalent to a TSP with 100 cities to visit. Figure 3-8 shows the evolution of a solution for the cross-shaped contour. The crossover probability was 0.6 and the mutation probability was 0.1. A solution appears at generation 78, and 100 solutions were obtained by generation 98.

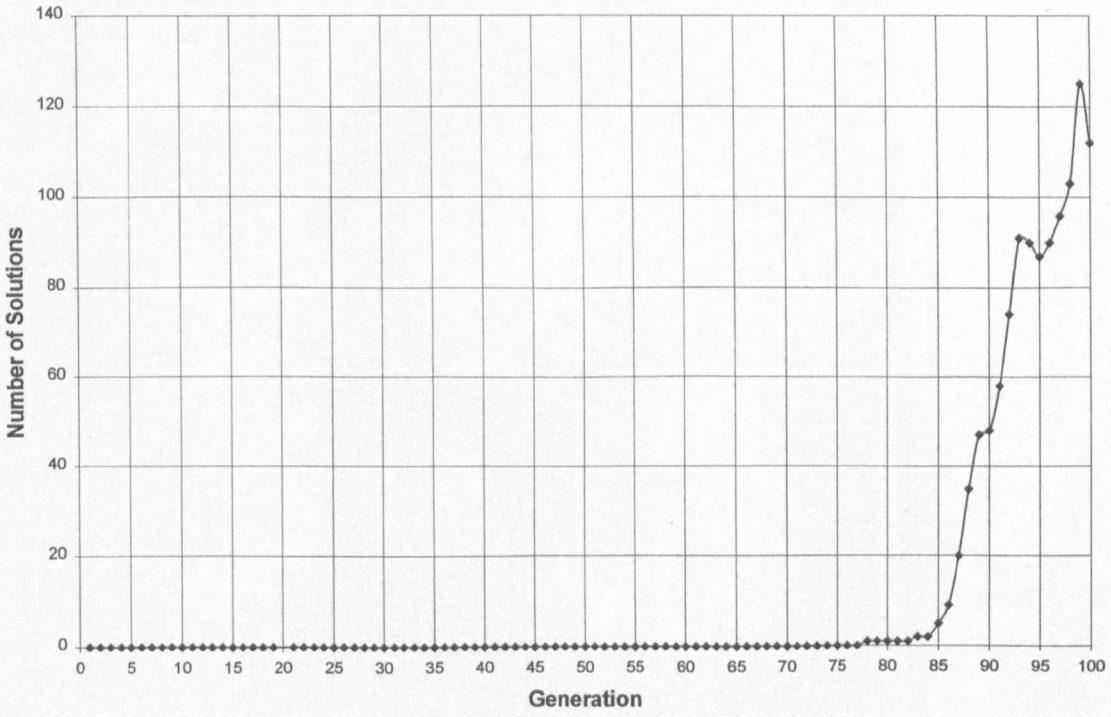

Figure 3-8: Solution Evolution for Cross Shape: Pm = 0.1

Figure 3-9 shows the evolution of the maximum fitness in each generation as the solution evolved. A 'best' fitness of 1.0 was achieved at generation 78, showing that the cross contour was correctly reproduced from the contour encoding information in Figure 3-5.

Figure 3-10 shows the evolution of the cross-shaped contour with the same crossover probability but with a mutation probability of 0.6. A solution appears at generation 61, with fitness 1.0 (Figure 3-11), that shows a correct reproduction of the cross shape. The number of solutions was less than for a mutation probability of 0.1 with oscillations in the number of solutions after generation 75. The larger value for the mutation probability appears to find a solution sooner but with more disturbance to the population later in the evolutionary cycle.

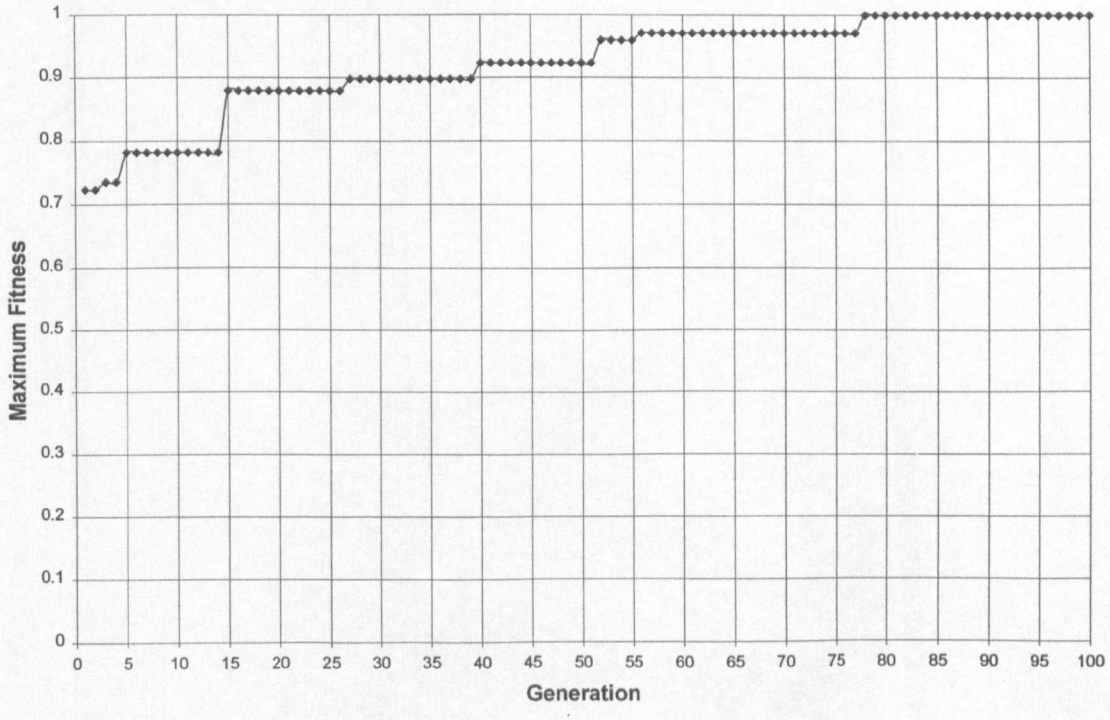

Figure 3-9: Population Maximum Fitness Evolution for Cross Shape: Pm = 0.1

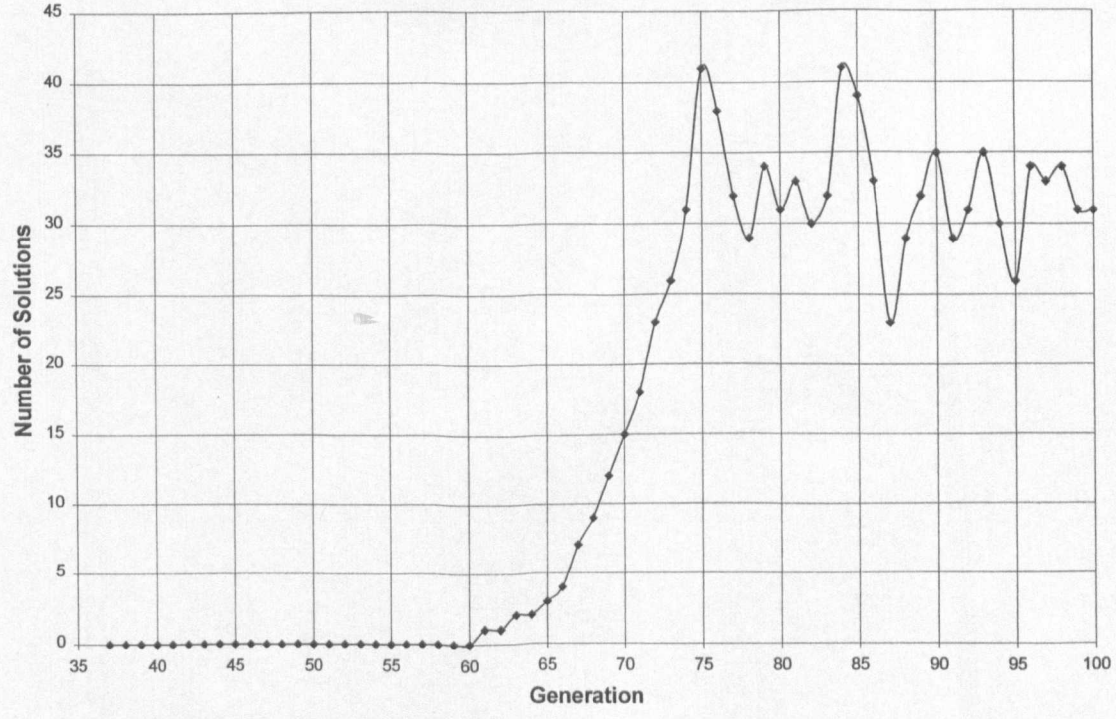

Figure 3-10: Solution Evolution for Cross Shape: Pm = 0.6

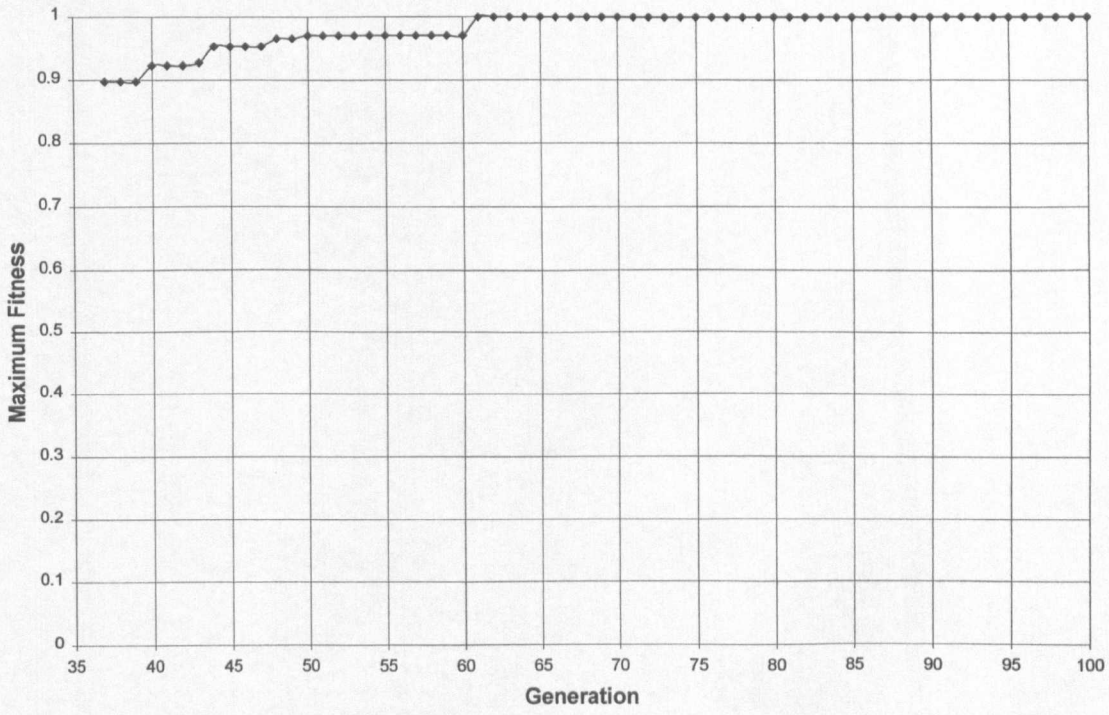

Figure 3-11: Population Maximum Fitness for Cross Shape: Pm = 0.6

The combination of the TSP and incorrect shape appearing from fittest cells indicated at this point in this study that the genetic algorithm could not solve the TSP (e.g. for the equivalent of 100 cities) with any reasonable success. The moment and Fourier descriptors, by themselves, could not provide a 'linear' fitness function. The problems associated with the calculation of the fitness for this type of process are further discussed in the research of Chapter 7. This form of contour encoding would, therefore, seem to be inefficient and difficult to use as an encoding format. A more suitable method is required, that is more efficient and easier to achieve.

3.6 Efficient Contour Encoding

Freeman (1961) described a method that permits the encoding of arbitrary geometric configurations in order to facilitate their analysis by a digital computer. It is shown that simple numerical techniques can determine whether a planar curve is open or closed. "Further, one can cause a given figure to be expanded, contracted, elongated, or rotated by an arbitrary amount." This paper

discusses the rectangular array type of encoding in which the contour slope function is quantised into a set of eight standard slopes. The paper noted that an encoding scheme for arbitrary geometric curves should be simple, highly standardised and universally applicable to all continuous curves. Freeman (1977) was concerned with the analysis of line drawings and defines a *line structure* as an assembly of points, line-segments and arcs in an n-dimensional Euclidean space. This paper also describes the various properties of a curve that is encoded with the eight-point *chain code,* such as chain inverse, chain length, closed chain centroid, area enclosed by a closed chain and polygonal approximation to a chain. A four-point chain code is also possible that will provide a coarser version of the contour co-ordinate information. Corner detection is also discussed and it is stated that "The presence of a sharp slope discontinuity in an otherwise smooth function is often an indication of an 'interesting' feature."

Lutz (1980) presented an algorithm for the analysis of thresholded digitised images. A thresholded image is composed of pixels that are either of type 'object' or 'non-object' and the problem of how to 'connect up' to the various object pixels is discussed. A series of touching object pixels is defined as a segment and the algorithm describes a single pass method for joining the segments together to form an object. Various rules are given that control the joining process and the use of a 'segment stack' forms the basis of the segment connection strategy. Capson (1984) also described a similar method and extracts a polygonal approximation to a contour in a raster-scanned binary image. This method operates on a single line of an image and execution is sequential in a single left to right pass in $O(n)$ time, where n is the number of transitions detected on a line. If the time to process one transition point is fixed, then the maximum time to process a line depends only on n and not on the topology of the objects in a frame. The algorithm described in this paper uses a dynamic data structure that is updated by simply changing a fixed number of links in the structure. There is no need to store the whole image and only the transition points of the raster line being processed are required. Memory storage must be provided for implementing the 'stack' type data structures and

for holding the list of contour co-ordinates. Object contours are linked counter-clockwise and hole contours are linked clockwise. In each case all collinear points are removed. The binary image is run-length encoded to provide the x co-ordinate of transition points between the object silhouette and the background in each scan line of the image. Post processing of the lists can determine information such as area, perimeter, moments, curvature and bounding box.

Wilson (1997) drew attention to the fundamental nature of the *crack coding* scheme and shows that the chain code and the mid-crack code may be derived from the crack code. Relationships are derived between simple measures of the contour perimeter and closed contour area. In this paper the contour is formed by traversing around the outer edge of every pixel on the edge of an object, i.e. along the cracks between adjacent pixels having complementary values. The direction of travel is such that the object pixels are on the right-hand side of the crack. "Where the object is an island the contour is traversed clockwise, while for a lake within an island the traverse is counter clockwise." A right enclosing octagon of a contour is introduced as an approximation to the convex hull, which is less sensitive to orientation than is the right enclosing rectangle. This paper suggests that crack coding is simpler to generate than other coding schemes since only four codes have to be generated rather than the eight required for chain coding. Estimates of contour area and higher moments are also simple to calculate.

3.7 Summary and Conclusions

As a result of the research experiments described in this chapter, it was considered that the minimal or most efficient coding produced by the genetic programming method, cannot be better than the Freeman chain code (see Freeman 1961 and 1977). The evolution of chromosomes that can be developed to follow a curve (genetic programming method) and also to reproduce the curve from the codes in the chromosome (TSP method) appeared not to be achievable from the experiments performed and described in this chapter. The curve or contour can be coded by various boundary

tracing processes and characterised by a set of features similar to those used for the fitness functions described above.

The research into the use of a genetic algorithm for providing the necessary shape features was thus redirected into investigating other ways to fully describe the 'dominant' features of a contour, so that improved recognition could be achieved and, if possible, include cases where the contour is partially occluded. The measurement of the dominant features on a contour will be discussed in Chapter 4, and the difficulties associated with the encoding of this dominant point information in a suitable manner for input to a genetic algorithm are investigated.

4. Perceptually Important Points on a Contour

4.1 Introduction

The research work described in Chapter 3 investigated the tracing of a curve or contour using a genetic programming algorithm. That method did not evolve chromosomes that would encode any particular features of the curve or contour. In order to investigate further the possible evolution of chromosomes that will encode information about the shape of a curve, the research developed a new method for encoding the various high curvature or Perceptually Important Points (PIPs) along a curve. The discussion of this new method stresses the suitability of the encoding for analysis by a chromosome in a genetic algorithm. The PIPs are the points along a curve, which have high curvature and can be considered to be important for the human observer to help in the recognition process. If a PIP encoding can be found that is suitable for analysis by a chromosome, then it should be possible to code the chromosome in such a way that some of the recognition information can be stored in the chromosome. A later recognition process can then use this information to help recognise a particular shape. The correlation of two contours is discussed and shown to have difficulties because of the phase discrepancies between the spatial frequencies of the contours. The normalisation of a contour before it is input to a genetic algorithm is also analysed. The form of the contour feature data, that is suitable as an input to a genetic algorithm, is an important concept that is developed in this chapter.

4.2 Review of Related Work.

A list of the related contour analysis references is provided in Figure 4-1. Most of the related work concerns the analysis of principal information points and dominant points on a contour. These references discuss a variety of ways to identify the points of high curvature on digitised contours. Active contour and multi-scale contour analysis research is also appropriate to the research work

described in this chapter. Little research work seems to have been reported in the literature on symbolic analysis or affine transform analysis. The Fourier and moment descriptor references in the table are of a general nature.

Related Work	References
Review Papers	Pavlidis, 1980; Trier, 1996.
Principal Information Points on a Contour/Curvature of Contour/Corner Detection	Brault and Plamondon, 1993; Chang, 1991; Chen, 1996; Cheriet and Suen, 1993; Dutta and Chaudhury, 1993; Fairney and Fairney, 1996; Fu, 1997; Gupta, 1995; Inesta, 1996; Koch and Kashyap, 1987; Koplowitz and Plante, 1995; Kovacs, 1995; Liu and Srinath, 1990; Pikaz and Dinstein, 1995; Saint-Marc, 1993; Sarkar, 1993; Sheu and Hu, 1996; Wang and Brady, 1995; Wu and Huang,, 1990; Xin, 1996; Yuan and Suen, 1995.
Dominant Point Analysis/Critical Point Detection	Cornic, 1997; Held, 1994; Inesta, 1998; Leaves, 1992; Tsang, 1994; Tsai and Chen, 1994; Verri and Uras, 1996; Yuen, 1994 and 1995; Yu and Yan, 1997; Zhang and Zhao, 1997; Zhu and Chirlain, 1995.
Fourier Descriptors/Moment Descriptors	Chen, 1990; Jiang and Bunke, 1991; Jia and Nixon, 1995; Kauppinrn, 1995; Li, 1993; Paulik, 1992.
Deformable Templates/Affine Transform Analysis	Cyganski and Vaz, 1995; Jain and Zongker, 1987.
Active Contours	Capson, 1984; Eviatar and Somorjai, 1996; Fejes and Rosenfeld, 1997; Kass, 1988; Park and Han, 1997; Xu, 1994.
Multi-Scale Contour Analysis	Asada and Brady, 1986; Bruckstein, 1997; Cesar and Costa, 1996; Cinque and Lombardi, 1995; Ip and Wong, 1997; Li, 1996; Marr and Hildreth, 1980; Mokhtarian and Mackworth, 1986; Mokhtarian, 1995, Ray and Ray, 1997; Tieng and Boles, 1997; Xin, 1996; Zhou and Pavlidis, 1994.
Symbolic Analysis	Olstad, 1991; Olstad and Torp, 1996; Pao, 1997; Saund, 1990.

Figure 4-1: Contour Analysis Algorithm References

Pavlidis (1980) reviewed a variety of algorithms for shape analysis and classified them under various criteria, such as whether they examine the boundary only or the whole area of the shape and whether they describe the original shape in terms of scalar measurements or through structural

descriptions. Detection of curvature maxima, combinations of curve fitting techniques, syntactic techniques and decomposition algorithms were among the methods reviewed. The integration of stochastic models and syntactic techniques and the further development of stochastic grammars were suggested as desirable methods. This paper noted that, in the long term, decomposition techniques may prove to be the most desirable, because they often provide descriptions which are closer to human intuition than the contour following techniques.

Trier (1996) presented an overview of feature extraction methods for off-line recognition of segmented (isolated) digit characters and noted that the selection of a feature extraction method was probably the single most important factor in achieving high recognition performance. The various feature extraction methods were discussed in terms of invariance properties, reconstructability and expected distortions and variability of the characters. The problem of choosing the appropriate feature extraction method for a given application was also discussed.

Brault and Plamondon (1993) defined the basic concepts associated with the measurement of a PIP along a contour. For each point on a contour, a vertex was iteratively constructed around this point, with the help of the neighbouring points either side (pairs), until some terminating condition is met. There must exist some geometric condition such that some pairs of points cannot be considered to be part of the *domain* of a vertex centred on that point. This paper also noted that pairs of neighbouring points do not contribute with equal weight to making a vertex look important.

Lin (1998) proposed a modified morphological corner detection method that found convex and concave significant points on a contour using simple integer arithmetic. Loss of corner information was reduced by a measurement of the line-segment distance between two contiguous significant points on the shape boundary contour. This method enabled the detection of concave and convex corners to be processed in parallel. Satisfactory results were obtained in detecting significant points that gave an accurate approximation to the shape boundary contour.

Sarkar (1993) described a simple, but efficient, algorithm for the detection of significant points of chain coded curves. A polygonal approximation was achieved by joining successive significant vertices.

Fu (1997) presented a Curve Bend Function (CBF) based method to extract the main features of a contour. This function utilised two quantities one was referred to as a Curve Bend Angle (CBA) that measured the bending degree at each point on the contour and the other that was referred to as a type coefficient that characterised properties such as convexity and concavity of simple curve segments. A local maxima or minima of the CBF corresponded to a critical point on the contour. The sign of a peak value indicated the related CBA being an inner or outer angle.

Wu and Huang (1990) developed a human face profile recognition system with learning capabilities. Cubic B-splines were used to extract the 'interesting' points (turning points) and a total of six interesting points are extracted. From the five segments of the curve, determined by these interesting points, twenty-four features were calculated.

Cinque (1998) presented a new method for shape description consisting of an approximation of a shape by a variable number of Bézier curve segments. Segments of the shape were approximated by using Bézier cubic curves which best interpolate the end points of the contour shape segments. Cubic polynomials were chosen because they were good approximations of the curve segments but were not as complex as higher degree polynomial functions. This method is translation, rotation and scale invariant. A Bézier segment was completely defined by the position of the two end points and the magnitude and gradient of the tangent vectors at these end points.

Yuan and Suen (1995) described an algorithm for identifying straight lines in a chain code representation of a contour. The algorithm determined the straightness of the digital arcs by constructing a passing area around the contour pixels.

Xin (1996) discussed contour matching in medical images and uses contour curvature as an important descriptor of a shape. This paper noted that curvature must satisfy the following requirements:

1. The curvature is invariant under rotation and translation.
2. The curvature is a local, scale dependent feature. A series of contours can be matched at any desired scale by using a multi-scale approach.
3. A curvature difference for corresponding points on two contours can be interpreted as the local deformation that has to be applied between these points in order to transform one curve into the other.

Zhu and Chirlan (1995) noted that a critical point is normally considered as a contour point with a maximum curvature. "For a parametric contour $x(t)$ and $y(t)$, if the second derivatives of $x(t)$ and $y(t)$ exist, curvature is strictly defined as:

$$k(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{\left(\dot{x}(t)^2 + \dot{y}(t)^2\right)^{3/2}} \qquad (4.1)$$

In this case, curvature at a point p is dependent only on the data in an arbitrary small region $(p-\delta, p+\delta)$. However, for a discrete curve, curvature can only be approximated. Approximation of curvature at a point p is dependent on the data in a support region $(p-k\Delta, p+k\Delta)$, where Δ is the sampling period. Because the support region is not arbitrarily small, the value of k must be found for every feature point. This is difficult, because different sizes of features require different support regions. This is why no strict mathematical definition of curvature exists for discrete curves."

Kass (1988) defined an active contour as an energy-minimising spline that was guided by external

constraint forces and influenced by image forces that will pull the active contour towards features such as lines and edges. Active contours locked onto nearby edges, localising them accurately. Active contours provided a unified account of a number of visual problems, including motion tracking and stereo matching.

Lai and Chin (1998) presented an integrated approach to modelling, extracting, detecting and classifying deformable contours in noisy images. This approach was based on a generalised active contour model (g-snake). The contour representation for an arbitrary shape was stable, regenerative and invariant under affine motions. This shape model was combined with Markov random fields that exerted influence over the arbitrary shape to allow for local deformations. Contour extraction was equivalent to an energy minimisation in the g-snake model.

Wong (1998) proposed a segmented snake (active contour) in which the global optimisation of a closed snake curve was converted into local optimisations on a number of open snake curves. A closed snake model was initially used to locate the contour near the object boundary where the overall energy reached its minimum value. A recursive split-and-merge procedure was then developed to determine the final object contour. Instead of using a constant normal force in the open snake, a function of the normal force along a boundary segment was measured based on a contour/not-contour type of criterion for the local contour segment. The proposed method was able to locate accurately all convex, concave and high curvature parts of an object boundary.

Pardo (1997) presented of velocity field estimation for the points on moving contours. This method determined the corresponding point in the next image by minimising the curvature change of a given contour point. This paper noted that, "Although the 3D motion of a contour changes the piecewise arc-lengths of the contour in a 2D image, the local curvature distribution along the contour is preserved relatively" and concluded that "Since the curvature distribution is preserved regardless of a contour motion, curvature information of the contour provides an important cue to

the point correspondence problem".

Asada and Brady (1986) introduced a representation of the significant changes in the curvature along the bounding contour of a planar shape. This representation was referred to as the *'Curvature Primal Sketch'* (CPS) because of the close analogy to the primal sketch representation advocated by Marr and Hildreth (1980) for describing significant intensity changes. A set of primitive parameterised curvature discontinuities was defined and an expression was derived for the convolutions of this set with the first and second derivatives of a Gaussian. An algorithm was described that calculated the CPS by matching the curvature discontinuities over a range of scales for a convolved shape.

Lee and Pan (1992) discussed off-line signature representations and proposed a multi-resolution critical-point segmentation method to extract local feature points at varying degrees of scale and coarseness. By representing a signature at varying degrees of resolution, the author was able to generate a hierarchical representation of the signature that can be used at a later stage for recognition and verification. A critical-point normalisation method was introduced to make the representation translation, rotation and scaling invariant. The normalisation method was based on the eigenvalues and eigenvectors of the covariance matrix of the distributed critical points.

Mokhtarian and Mackworth (1986) discussed the problem of finding a description, at varying levels of detail, for planar curves and matching two such descriptions. Path-based Gaussian smoothing techniques were applied to the contour to find zeros of curvature at varying levels of detail. The result was the 'generalised scale space' image of a planar curve that was invariant under rotation, uniform scaling and translation of the contour. Mokhtarian (1995) described a light box that is used to obtain silhouette images that are segmented to obtain the boundary contour of an object. The boundary contour was classified as convex or concave. Convex curvature was recognised using

four high-scale curvature extrema points. *Curvature Scale Space* (CSS) representations were calculated for concave curves. The CSS model was a multi-scale organisation of the natural, invariant features of a contour (curvature zero-crossings or extrema) and useful for reliable recognition of the correct model because it placed no constraints on the shape of an object.

Cong and Ma (1998) addressed the problem of corner enhancement based on the principles of curvature scale space. A General Geometric Heat Flow (GGHF) algorithm was used to investigate the conditions that satisfy the scale space criteria. Corner enhancement criteria were also proposed which required that new corners should not be generated as the existing corners were enhanced. A new curve evolution scheme was presented that can enhance strong corners, suppress noise and was able to satisfy the scale space criteria.

Rosin (1998) presented an alternative method to representing curves at a single or fixed number of scales that represented a curve at its *natural* local scales. Several approaches to determining the local natural scales of curves were described. This paper noted that various, possibly overlapping sections of curves, should be represented at certain specific scales depending on their shape. Results were presented for the zero-crossing of curvature density method, with smoothing applied to merge similar curved sections.

These extracts from the relevant literature indicate a selection of the techniques used to identify the major shape characteristics of a contour. The new technique for identifying the PIPs along the length of a contour that is developed in this chapter uses similar principles as the techniques discussed above. The main function of the new technique is to collect the contour feature data in such a way that this information can be input into a genetic algorithm is a satisfactory manner. The collection of curvature information was chosen because it can take into account a varying resolution of data collection along the curve and, using the spatial frequency normalising method, the curve can be filtered to provide multi-scale information. Thus the contour can be analysed with a fine and

coarse resolution which will potentially increase the information available for recognition purposes.

4.3 Contour Analysis and Normalisation

4.3.1 Image Contours

Images have contours of some feature, usually grey level. The contour can be of other features such as a first differential or edge filtering operation or the zero crossings of a second differential filtering operation. Assuming any particular contour can be segmented from the other contours a method has to be defined to filter the contour in such a way that the principal information or curvature points (PIPs) can be measured. The basic information about a contour can be obtained as a chain code, which stores the directions of the curve in units of 45 degrees (see Chapter 3). The contours may be closed or touch the side of the defining patch. The shape characteristics or features can be contained in either the contour or the region enclosed by the contour. Only the shape contour PIP information was considered in this research and closed contours were preferred because the Fourier theory, which was used, applies strictly to cyclic data. Contours touching the defining patch boundary can be tagged as such and sections on the patch boundary will be expected to be 'straight'. Contours may also be 'inside' or 'outside', depending whether the boundary tracing was along the outside of the contour (clockwise) or along the inside of the contour (anti-clockwise).

4.3.2 Contour Normalisation

The contour information has to be normalised in some manner, in preparation for the recognition part of the process. A shape contour can be at any orientation, at any position in the image and the object may be of any size relative to the size of the image. Various views of the contour have to be adjusted to a normalised orientation, position and size before any recognition processes are applied. Fourier theory was used to perform this normalisation.

This research extrapolated the x, y co-ordinates of the contour to fill the nearest larger 'power of

two' array. This interpolation is a simple sub-sampling of the original curve. The contour co-ordinates were then transformed to produce the spatial frequencies associated with the contour. The contour co-ordinates have to be sampled and stored into the real and imaginary parts of the input data array. The Discrete Fourier Transform (DFT) is made invariant to translation by setting the zeroth spatial frequency, which is equivalent to the centre of gravity or centroid of the contour, to zero. The centroid is recorded for comparison with other contours when objects are expected to have more than one contour associated with them.

The principal axes for the first spatial harmonic were calculated and the spatial frequencies were rotated by the angle that the major axis makes with the horizontal (x-axis). A check was applied to make sure the major axis is aligned with a standard direction. The addition of 180 degrees is sometimes required to normalise the direction of the major axis of the contour. The spatial frequencies are now normalised for rotation in the plane of the 2D contour. The spatial frequencies now have to be normalised for size. This can be achieved by dividing all spatial frequencies by the modulus of the first harmonic. The normalisation process must also preserve the correct aspect ratio of the object.

The final normalisation required that the starting point be adjusted to line up with the correct end of the normalised major axis of the first harmonic of the spatial frequencies. Each spatial frequency was rotated by an amount proportional to its frequency. The various methods for achieving this correction are discussed below. The literature survey indicated that this starting point correction is often not performed but it is essential for improved results when using a contour for comparison with a training set contour database for classification purposes.

The fully normalised spatial frequencies, together with their various normalisation parameters, can now be stored as a representation of the original contour. Typically the first, say, 30 spatial frequencies, were recorded so that further processing is not too excessive. Note that the spatial

frequencies have to be stored for both the x and the y co-ordinates and that each spatial frequency is a complex number.

4.3.3 Related Work on Fourier Descriptors

The Fourier spatial frequencies or Fourier Descriptors (FD) themselves can be used to classify various contours into their respective classes. Mahmoud (1994) used Fourier descriptors and contour analysis for the recognition of Arabic characters. The contours of the primary part of the character were extracted, together with the dots and the hole contours. The Fourier descriptors and contour curvature features were calculated for the primary part of the character. The dots and hole features were then used to further identify a character. The features are compared to a model using a distance measure. The run-lengths and direction of the contour chain codes were also used in the recognition process and are classified into concave and convex features. This paper noted that, "... the use of several recognition techniques is necessary to achieve higher recognition rates for Arabic characters".

Wu and Sheu (1997) used Fourier descriptors to achieve contour correspondence between two stereo images. The contours were expressed as epipolar lines of which the slope and the intercept functions were expressed as Fourier series. The contour correspondence was calculated using a Minimal Spectrum Distance (MSD). This paper noted the following:

1. Fourier descriptors are capable of approximating functions with finite discontinuity.
2. The constant term of the Fourier descriptors gives the position of the contour.
3. The first harmonic term associated with the Fourier descriptors gives the orientation of the contour.

4.3.4 Related Work on Moment Descriptors

The techniques of moment analysis can also be used to further clarify the FD, by adding particular extra features to those provided by the Fourier descriptors. Teague (1980), Cash and Hatamain

(1987), Jiang and Bunke (1991), Mukundan (1993), Wang (1994), Sardana (1994), Wallin and Kuber (1995) and Slusek (1995) provided good descriptions of the theory and application of moments to the problem of obtaining good contour or shape features that are suitable for shape recognition.

Jiang and Bunke (1991) addressed the problem of efficient computation of moments from the boundary of a digital area and considered that boundary-based computation was better than the usual region-based approaches, because the data dimension of boundary representations was substantially smaller than that of region representations. A simple iterative algorithm was proposed for the calculation of moments from a polygonal approximation of the boundary contour. An adaptation of the algorithm can also be used to calculate the moments from the run-length chain code of the boundary.

Mukundan (1993) considered the problem of estimating the 3D rotational parameters of a rigid body from its monocular image data, and used moments to identify the camera view axis direction in the body-fixed reference frame. The relative attitude of the rigid body was then expressed in terms of quaternion parameters to model the outputs of a video sensor in attitude control simulations.

Sardana (1994) investigated the use of internal edge details of an object, especially when the information does not form a closed curve. Edge Standard Moments (ESM) were developed that are invariant to location, scale and rotation. Invariant moments were usually applied to the silhouette of an object and this paper extended this concept to the case of arbitrary edges.

Wang (1994) proposed a new shape descriptor, called Moment Fourier Descriptor (MFD), that can describe a complex object composed of a set of closed regions. This descriptor was shown to be independent of the translation, rotation and scaling of the object. The essential advantage of this

descriptor was that it could be used to recognise more complex patterns than the traditional Fourier descriptors. After the area centroid was found and the number (N) of angularly equi-spaced radial vectors were formed, the moments of every radial vector were calculated. By performing Fourier analysis on these moments the moment Fourier descriptors were obtained.

Mertzios and Tsirikolias (1993) referred to the use of a modified set of moments for the development of efficient shape discrimination and classification. These moments were normalised with respect to standard deviation of the contour pixel co-ordinates and appear to improve classification performance. They were also less sensitive to noise.

Slusek (1995) described a prototype object by a family of shapes. These shapes were created by occluding the object by circles (of different radius) located at the centre of the object area. The moment invariants of such shapes were functions of a parameter describing the size of the circles. Using these functions it was possible to create many shape descriptors from a single moment invariant. There was, therefore, no need to use the higher order moments. This paper noted that moments of higher order than two are sensitive to digitisation errors, minor shape deformations, camera non-linearity and the non-ideal positioning of the camera. A method was presented for using moment invariants of order two, to create more position invariant descriptors and to improve the resolution of these descriptors.

4.3.5 Moment and Fourier Descriptor Problems

The problem usually remains that a variety of different shapes have very similar Fourier descriptor and moment descriptor values, especially when an *rms* Euclidean error method is used. This particular problem also manifests itself when moment descriptors or Fourier descriptors are used as a fitness measure for a genetic algorithm. As mentioned in the research of Chapter 3, the fitness of a chromosome may be improving and indicating that a certain shape comparison is the fittest for a particular generation, but another chromosome can be reproduced that apparently has a better shape

comparison depending on the *rms* error method being used. When the human observer compares the two shapes the second chromosome's comparison is in fact worse than that of the first chromosome. Hence a further 'description' of the contour is required.

The Fourier descriptors are often used in the spatial frequency domain where the phase information is ignored by using the magnitude only of the complex spatial frequency, or the phase information adds too much variation to the shape characteristics, so that recognition success is limited. In general it is preferable to work in the spatial domain, in order to reduce the interference from the phase information in the spatial frequencies and to collect 'spatial' features of the contour for an input to a genetic algorithm. Sakano (1996) discussed the reconstruction of a contour from feature vectors, and highlighted the difficulty of finding the best operators for the reconstruction. The solution may not be unique, hence the loss of recognition capability. A proper matching of the contour information to the genetic algorithm for analysis is by no means trivial, and will be discussed later in this chapter. The stored Fourier descriptors and any associated parameters (e.g. moment descriptors, aspect ratio, and relative centroid) are used to reproduce the original contour in the x-y plane.

4.4 Contour Correlation

This section describes several experiments designed to show the difficulties and problems associated with the correlation process. Figure 4-2 to Figure 4-9 show the first five spatial frequencies of a variety of contours and the correlation of the radial spokes (Ip and Shen, 1998) for the same contours. Radial spokes are drawn from the centroid to the perimeter of the shape contour at equal angle intervals. The length of the spokes forms the basis for the calculation of features for the contour. The number of spokes is typically 60, but depends on the spatial resolution required for the calculation of the contour features. Figure 4-2 shows the spatial frequencies for two normalised aircraft contours that are similar. The aspect ratios (0.49 and 0.38) are for the first spatial frequency

ellipse of each shape respectively. The ellipses show the phase differences between the x and y spatial frequencies. Even for two very similar contours, it can be seen that the magnitude and phase of the spatial frequencies differ significantly. The two lower graphs, in Figure 4-2, show the *rms* error between the x and y magnitudes of the various spatial frequencies. The *rms* error is plotted vertically and the number of spatial frequencies is plotted horizontally. The first *rms* error in the x spatial frequency is zero, because the spatial frequencies are normalised to give a value of 1.0 to the first x spatial frequency magnitude. The maximum x_{rms} magnitude error is obtained as 0.069 for the second spatial frequency and the maximum y_{rms} magnitude error is 0.056 for the first spatial frequency.

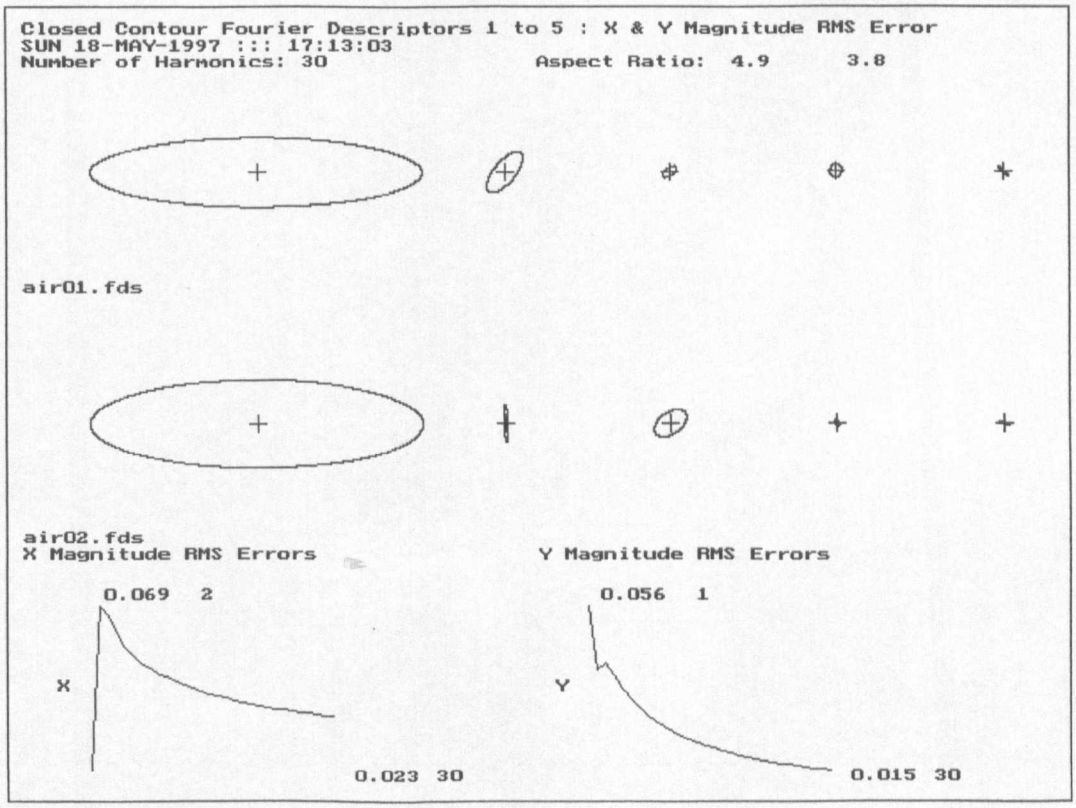

Figure 4-2: Aircraft Fourier Descriptors 1 to 5

Figure 4-3 shows the two contours in the x-y plane as transformed back from the spatial frequency domain. Thirty spatial frequencies have been used for the transform. Superimposed on the contour is an equivalent contour, produced by calculating the radial spokes from the centroid of the contour at six degree intervals (Angular Bin Size = 6). A histogram is produced of the radial spokes at the

six-degree angle intervals and the longest spoke in the histogram bin is selected for the display. Increased resolution of the angular bin size will allow the spokes to match the contour more accurately, but will require more processing time.

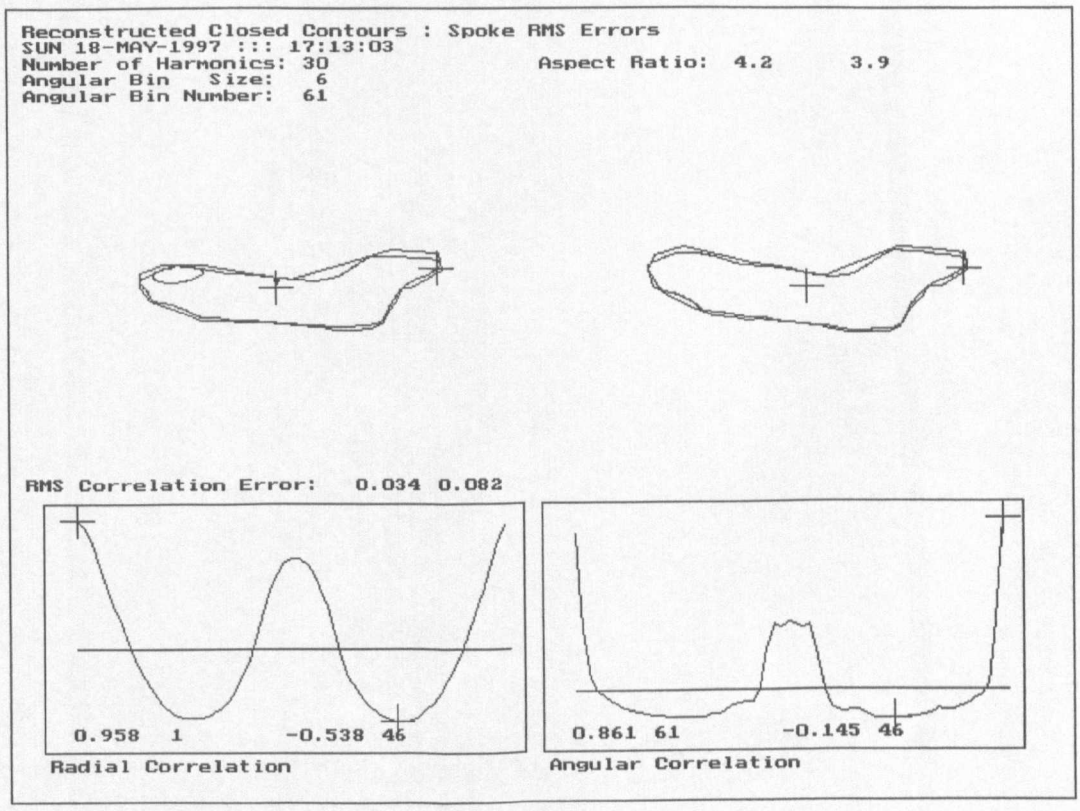

Figure 4-3: Aircraft Spoke RMS Errors

A one dimensional (1D) correlation is performed on the length and angle of the longest spoke in each histogram bin. The shape of the correlation output curve is shown in the lower half of the display (Figure 4-3). The value and position of the maximum and minimum in the correlation output are also displayed and are identified by a cross on the output curve. For these two similar contours, the maximum radial correlation is 0.958 at position 1 in the histogram. The maximum angular correlation is 0.861 at position 61 in the histogram. These values indicate good correlation, as would be expected. Comparison of the *rms* magnitude error also shows a small value, thus confirming the good correlation results.

Figure 4-4 and Figure 4-5 show the results for the correlation of a contour with itself. As expected,

the *rms* magnitude error for the thirty spatial frequencies is zero and the correlation of the spoke lengths and angles is 1.0 at position 1 in the histograms.

This contour example is the author's signature, which has been reconstructed using thirty spatial frequencies.

Figure 4-6 and Figure 4-7 display the spatial frequency and correlation results for two similar contours of the digit two. The left digit was hand written by the author and the right digit was obtained by scanning the image database for the Open University tutorial, T396 Block 3 Technology, pp. 8 - 9. The x_{rms} magnitude errors are low when only including the first two spatial frequencies but the y_{rms} magnitude errors are quite large at 0.252 in histogram position 1. The radial correlation is high, (0.758 at histogram bin position 1), but the angular correlation is low, being only 0.462 at bin position 2. Both the *rms* magnitude error and the correlation information, for these two contours, show how sensitive the correlation process is to these measures or features of the contours.

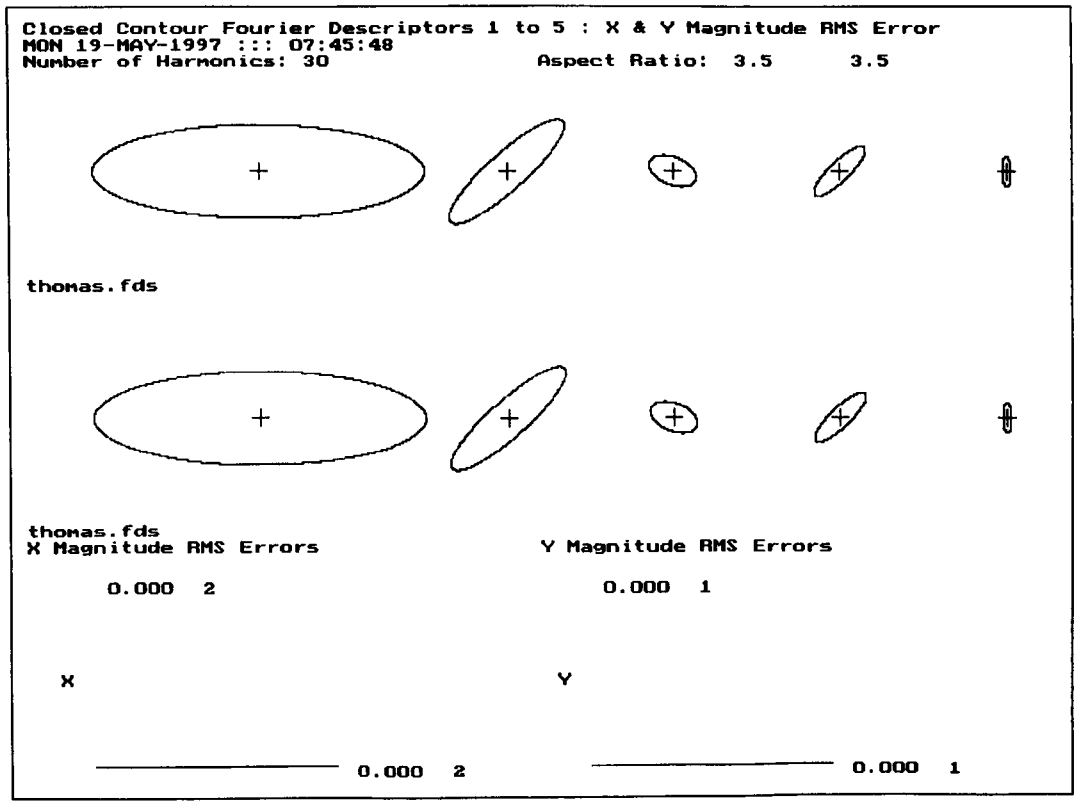

Figure 4-4: Signature Fourier Descriptors 1 to 5

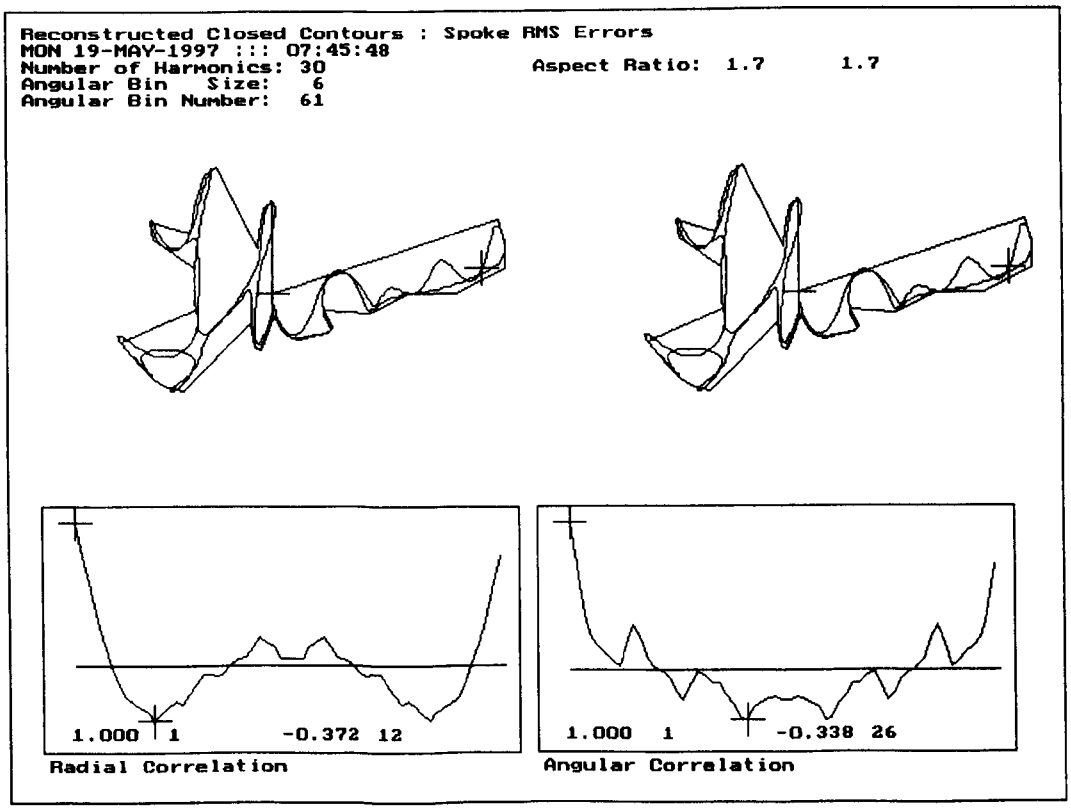

Figure 4-5: Signature Spoke RMS Errors

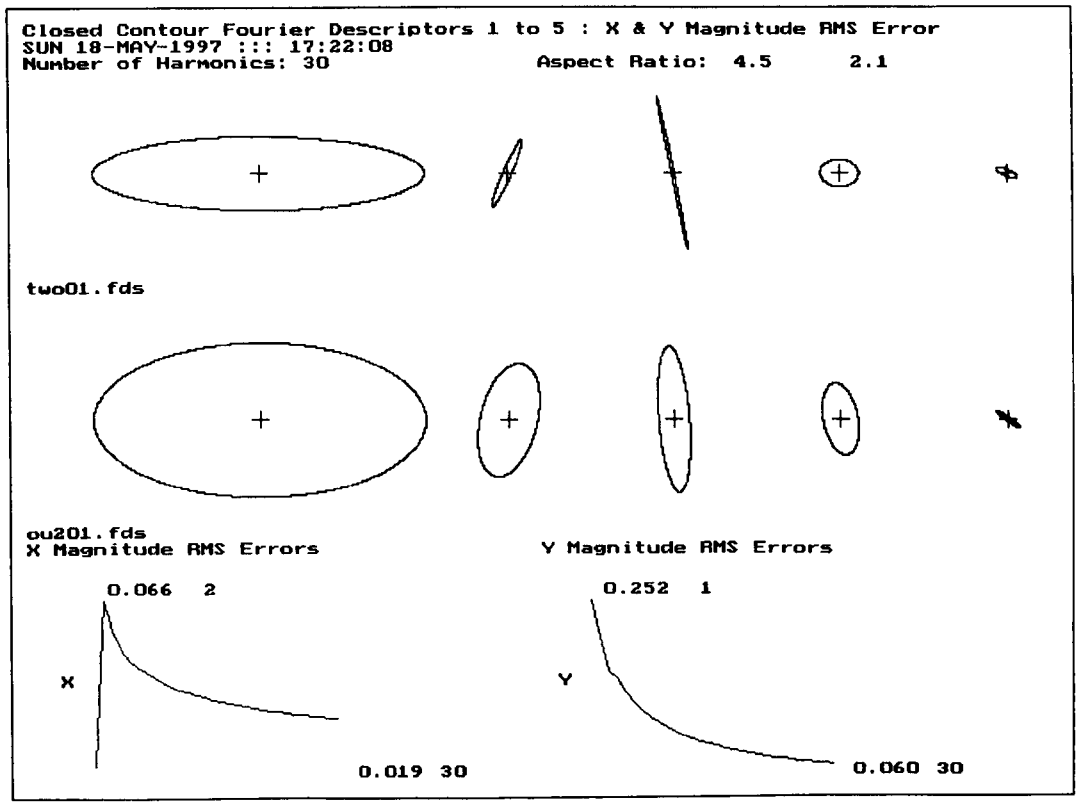

Figure 4-6: Digit Two Fourier Descriptors 1 to 5

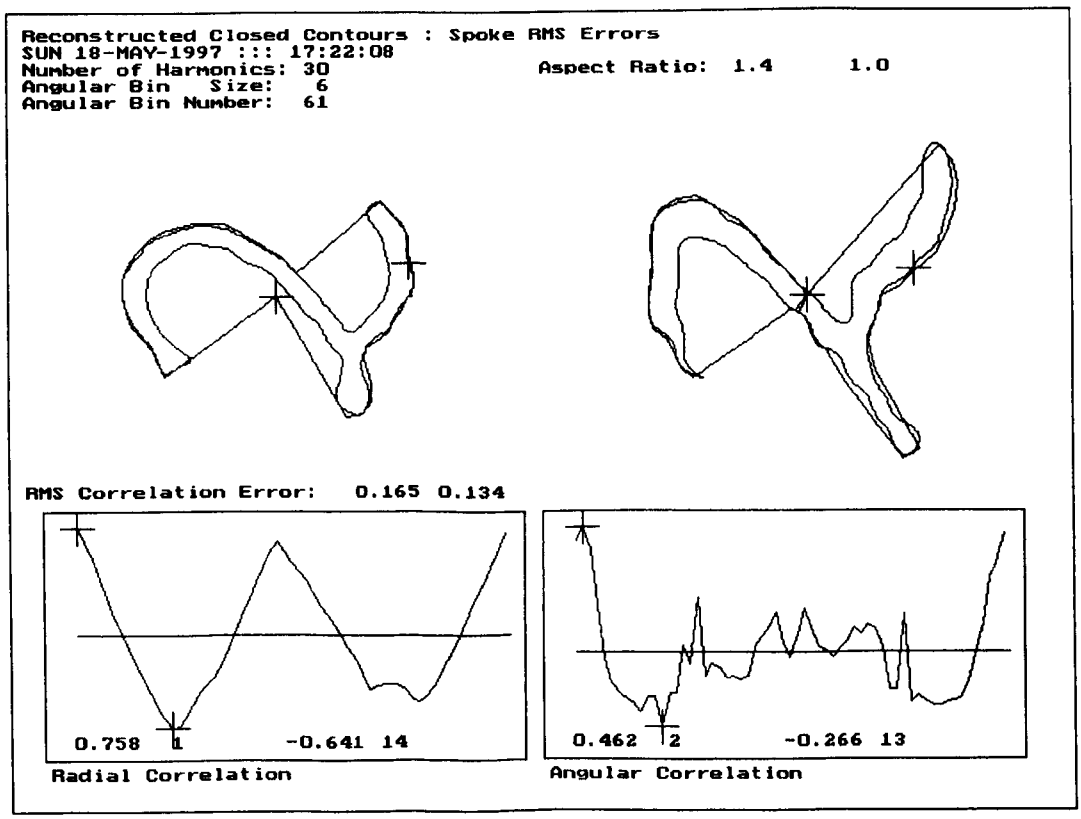

Figure 4-7: Digit Two Spoke RMS Errors

Figure 4-8 and Figure 4-9 show the features for two dissimilar contours: an aircraft and the digit

two. The *rms* magnitude errors indicate high values, (0.302,) only in the y direction. The radial and angular correlation results show a poor match, 0.578 at bin index 29, for the radial histogram and, 0.312 at bin index 28, for the angular histogram. In this case, the correlation information is required to be able to distinguish the two contours. The *rms* magnitude errors would be inconclusive.

The *rms* correlation error shown above the radial correlation curves on the figures indicates the *rms* error between the radial and angular correlation curves shown in each figure. Small values would indicate a good match between the two correlation curves. Figure 4-3 shows good matching with values of 0.034 and 0.082. Figure 4-5 does not display any values for the case where the two contours are the same. Figure 4-7 shows larger values than Figure 4-3, (0.165 and 0.134,) even though the contours are similar. Figure 4-9 gives similar values to Figure 4-7 but with two dissimilar contours. It would seem, therefore, that the similarity of the 1D radial and angular correlation curves is not necessarily a good indication of the similarity of the original contours.

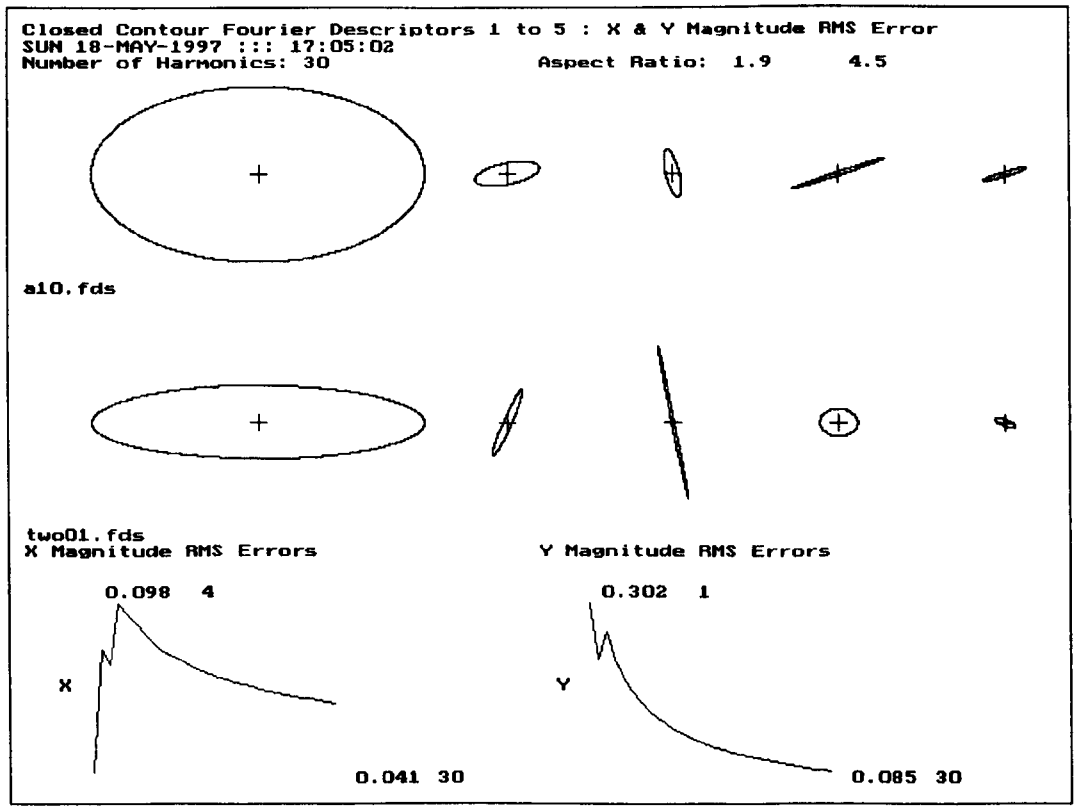

Figure 4-8: A10 Aircraft Fourier Descriptors 1 to 5

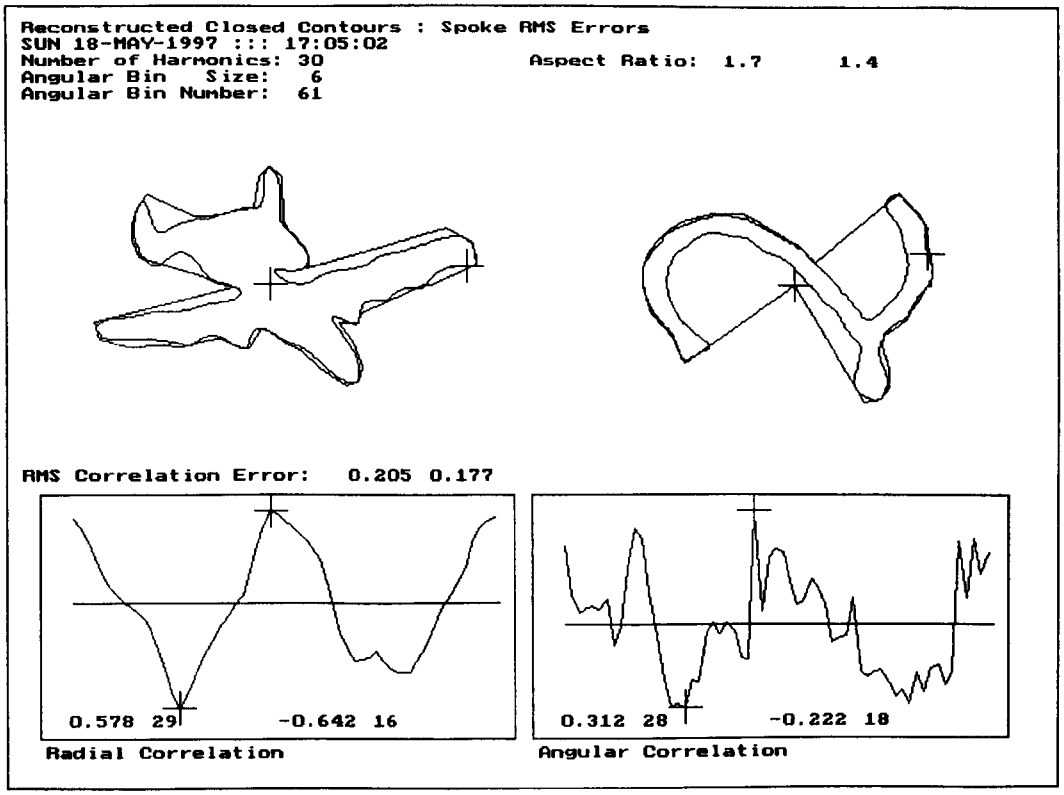

Figure 4-9: A10 Aircraft Spoke RMS Errors

These research results have been included to indicate the problems that occur when Fourier descriptors are used as features of a contour. The *rms* magnitude errors normally show large differences when contours are dissimilar, and the differences for contours that are similar but of different shapes are usually small and can vary considerably. The *rms* magnitude errors do not, in general, provide enough resolution to distinguish similar but different shapes. The 1D correlation of the radial and angular information of a contour usually provides better results. Because the spokes are measuring the radial dimensions of the contour, the correlation curves are dependent on the methods used to normalise the contour information. Hence a further variety of matching methods are required to help recognise a particular contour, in order to avoid a variety of objects being matched as being similar which the human observer would say were not the same.

A further measurement on a contour can be made which will improve the discrimination between

similar shapes and, at the same time, not match dissimilar shapes. The principal curvature positions or PIPs along a contour can be measured and used to provide a 'description' of the contour in terms of its major 'shape characteristics'. These measurements are discussed later in this chapter in section 4.6.

4.5 Contour Fourier Analysis

4.5.1 Introduction

The various methods of Fourier analysis and the new PIP calculation technique are now described in detail. Different parts of the contour can be allocated a particular 'description' that can be used to refine the previously mentioned ambiguities. A form of multi-scale (resolution) analysis can also be provided, when ambiguities have to be resolved. The multi-scale can either be in the form of an 'increasing length window' that is used to measure the PIPs, or the spatial frequency bandwidth used to reconstruct the contour. The identification of parts of a contour may be possible, which may introduce the possibility for the recognition of partially occluded objects.

The following textbooks describe the relevant properties of the Fourier descriptors, Gonzalez and Wintz (1987, pages 404 to 409), Dougherty and Giardina (1988, pages 370 to 385), Heijden (1994, pages 261 to 270) and Sonka (1994, pages 205 to 212). Zahn and Roskies (1972) provided a general reference for the Fourier descriptors of a plane closed-curve. Rosin and Venkatesh (1993) suggested that "... contours contain many different sized structures and hence these structures need to be described at multiple scales", and "... instead of describing the contour at all scales, it is more efficient to identify the most significant scales that best represent the structures in the contour". Granlund (1972), Pei and Lin (1995) and Rothe (1996) all considered the various methods for the normalisation of a contour. Granlund (1972) noted that the Fourier coefficients contain information about size, orientation and a phase factor that is dependent upon the starting point along the contour.

Pei and Lin (1995) developed a normalisation algorithm that transforms a pattern into its normal form, such that it is invariant to translation, rotation, scaling and skew. The covariance matrix of a pattern was first calculated and the pattern was then rotated according to the eigenvectors of this covariance matrix. The pattern was then scaled along the two eigenvectors according to the eigenvalues, in order to bring the pattern to its most compact form. After this process, the pattern was invariant to translation, scaling and skew. By applying tensor theory, a rotation angle was found which could make the pattern invariant to rotation.

Rothe (1996), in particular, discussed the affine invariants that have not been included in the contour normalisation process described in this chapter.

The Fourier analysis of a contour forms an essential part of the research work described in this chapter. The analysis is used for two purposes: firstly to filter the spatial frequencies of the contour, e.g. low pass filter the contour, and secondly, after normalisation for translation, rotation, size and start point, to reconstruct the curve in the spatial domain for shape recognition purposes. The extraction of features takes place in the spatial frequency domain and in the spatial (x-y co-ordinate) plane. The spatial frequency domain, by itself, is not sufficient to fully distinguish one contour from another. Information from the spatial domain is also required, especially when part of the contour may be obscured.

4.5.2 Fourier Analysis Principles

The main principles of Fourier analysis, as described in the references and text books, are included for completeness and are listed as follows:

1. The boundary of a region is an unambiguous representation of the region.
2. Any parameters that describe the boundary implicitly characterise the region.
3. The boundary becomes more descriptive if the representation is that of a closed curve.
4. The contour of a region refers to an ordered set of boundary points.

5. Let (x_0, y_0) be the start point for the contour. The length of the perimeter is denoted by P. Let s be the running arc length of the boundary traversed in the clockwise direction. The traversal ends at the point (x_0, y_0), when all the boundary points have been visited, and where $s = P$.
6. The contour associated with a region is a path consisting of the boundary points (x, y) of the region.
7. There are two main representations of the contour, namely: Freeman chain codes (Freeman, 1961 and 1977) which describe the direction of the contour at each point and a boundary list, an ordered list of (x, y) co-ordinates for the contour.
8. The contour points can be either 4-way or 8-way connected. 4-way connected points trace a path that only has the four compass directions North, East, South and West. 8-way connected points may be joined diagonally as well (NE, SE, SW and NW). Care must be taken when sampling the contour that the samples are equally space in the spatial domain.
9. The boundary list corresponds to a parametric curve $[x(s), y(s)]$, where s is the running arc length relative to the start point (x_0, y_0).
10. The co-ordinates $[x(s), y(s)]$ can be regarded as two periodic functions with period P.
11. Since the contour is closed, i.e. no gaps, these functions are continuous.
12. The image plane can be considered as a complex plane such that

$$z = x + jy, \text{ where } j = \sqrt{-1} \tag{4.2}$$

13. The closed contour is now a complex, periodic and continuous function, i.e.

$$z(s) = z(s + P) \tag{4.3}$$

14. Fourier series expansion of a periodic function requires the calculation of the complex amplitudes Z_k of harmonic functions, with frequencies that are multiples of $\left(\frac{1}{P}\right)$.

15. Therefore we can write:

$$z(s) = \sum_{k=-\infty}^{+\infty} Z_k \, esp\left(\frac{2\pi jks}{P}\right) \tag{4.4}$$

and

$$Z_k = \frac{1}{P}\int_{s=0}^{P} Z(s) esp\left(\frac{-2\pi jks}{P}\right) ds \tag{4.5}$$

where $k = \ldots -2, -1, 0, +1, +2, \ldots$

16. These complex harmonics (spatial frequencies) are named 'Fourier descriptors' of the contour.

17. For a uniformly sampled contour, the Fourier descriptors can be calculated using the DFT:

$$Z_k = \frac{1}{N}\sum_{n=0}^{N-1} Z_n \, esp\left(\frac{-2\pi jnk}{N}\right) \tag{4.6}$$

where Z_n are the sampled contour points and N is the number of contour points.

18. When the number of sample points is a power of 2, the more efficient Fast Fourier Transform (FFT) can be used.

4.5.3 Properties of Fourier Descriptors

The various properties of the Fourier descriptors are listed for completeness as follows (these properties could also be used as part of the feature set for a contour):

1. If the contour is non-fractal like, the perimeter must be finite. A finite number of terms of the Fourier series can establish an approximation of the contour with arbitrary accuracy.

2. The area (A) of the contour is given by:

$$A = \pi \sum_{k=-\infty}^{+\infty} k \left| Z_k \right|^2 \tag{4.7}$$

3. The perimeter (p) of the boundary is given by:

$$p^2 = 4\pi^2 \sum_{k=-\infty}^{+\infty} k^2 |Z_k|^2 \tag{4.8}$$

4. Normalisation for translation. Z_0 is the centroid of the contour, but not necessarily of the region. This term is set equal to $(0,0)$ to normalise the spatial frequencies.

5. Normalisation for scaling or size. When the whole contour is scaled by a factor *S*, all Fourier descriptors are scaled by the same factor *S*. Hence normalisation is achieved by dividing all Fourier descriptors by modulus $[Z_1]$ (First spatial frequency). The aspect ratio of the contour should be preserved by normalising the *x* and *y* co-ordinates separately, otherwise rectangles will not be distinguished from squares. This normalisation effectively makes the first harmonic ellipse for all shapes approximately the same size (i.e. the major and minor axes of the first harmonic ellipses are the same lengths).

6. Normalisation for rotation. When a contour is rotated by an angle θ, all Fourier descriptors are multiplied by $e^{j\theta}$. Note this will be a complex multiply in the spatial frequency domain. The angle of rotation is obtained from the first harmonic ellipse, and is the angle of rotation which is required to place the major axis of the first harmonic ellipse parallel to the horizontal *x*-axis, i.e. zero phase angle. An angle of 180 degrees may have to be added to the rotation angle in order that the directions of the major and minor axes are all normalised correctly.

7. Normalisation for start point. The start point for the contour will depend on the original orientation of the contour. In order to normalise the contour for a varying start point position, the start point is moved an angle Φ given by $\Phi = \frac{2\pi s_0}{P}$ where s_0 = distance along the contour arc which places the start point at the end of the major axis of the first harmonic. In order to adjust correctly the phases of all the other harmonics, each harmonic is multiplied by $e^{jk\Phi}$, where *k* is the harmonic frequency, i.e. each harmonic is rotated in the spatial domain, relative

to the first harmonic, by an amount that is proportional to its spatial frequency. The spatial frequencies of the contour will then 'line-up' and reconstruct the correct shape for the contour.

8. Only the spatial frequencies of interest for a particular contour need to be processed. Once the spatial frequencies of the contour have been normalised, an inverse DFT can be performed to produce a normalised contour. The spatial frequencies are now in units of cycles/contour and, therefore the number of points on the reproduced contour can be chosen to suit the resolution required by the problem. Thus all normalised curves will have the same number of points, which will make comparisons between curves more accurate. The number of contour points used in the following analysis was chosen to be 512, because this value was between the lowest sampling value, 128, and the highest sampling value, 2048, used to digitise the test contours. The normalised contour can now be used to collect distinguishing features of the shape. Figure 4-10 shows an aircraft reconstructed from 200 spatial frequencies, Figure 4-20 shows the same aircraft with 10 spatial frequencies and Figure 4-29 shows the aircraft contour reconstructed using 25 spatial frequencies.

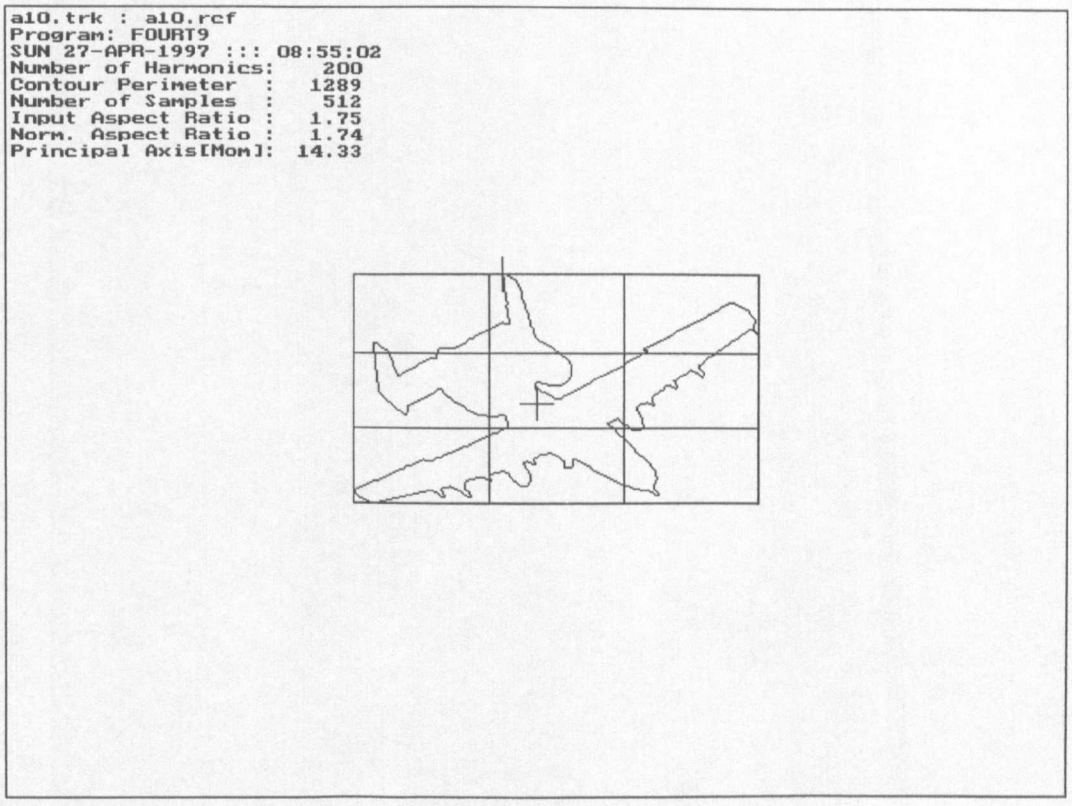

Figure 4-10: Aircraft Contour : 200 Spatial Frequencies

4.6 Contour Perceptually Important Point Measurement

The research work described in this section discusses the problem of how the PIP measurements on a contour can be input into a genetic algorithm in a suitable manner. The usual use of a genetic algorithm is to adjust the parameters associated with the solution(s) of a problem. The PIP data collected by this new method described below has been developed specifically during the research for genetic algorithm input and not primarily as a PIP calculation technique.

The contour points are available as (x, y) co-ordinates and/or Freeman direction chain codes (0 to 7). The Freeman chain codes represent the directions clockwise in steps of 45 degrees, with 0 indicating a North direction and 4 indicating a South direction. The contour is obtained from the output of the Fourier smoothing and normalisation process described above. The number of contour points, N, will be of a constant value (512). This new method is similar to those proposed in the literature, but is significantly different because of the emphasis on collecting information suitable

for input into a genetic algorithm.

The important characteristics of this new technique for calculating PIPs along a contour can be summarised as follows:

1. Use is made of the efficient Freeman chain code.
2. A coarse/fine sampling of the contour is provided.
3. The use of Fourier analysis for the normalisation of a closed contour allows for multi-scale processing of the contour.
4. The technique can be satisfactorily applied to geometric shapes.
5. The implementation is simple and is suitable for a hardware implementation.

The PIPs on a contour are calculated as shown in sections 4.6.1 to 4.6.4

4.6.1 Collection of Direction Histogram

A direction histogram is collected for each point on the contour. The direction histogram is an 8 by 8 matrix that identifies the contour change in direction (curvature) at each point along the contour. The histogram position $[j,k]$ is incremented such that $[j]$ is the direction (0 to 7) at a point (i) on the contour, and $[k]$ is the direction (0 to 7) at a point $(i+1)$ on the contour, where (0 to7) are the Freeman chain code directions. The 8 by 8 matrix allows for both 4-way (using the 0, 2, 4, 6 Freeman chain code directions) and 8-way (using the Freeman chain code directions 0 to 7, (see also Chapter 3)) connected contour points. The direction histogram is shown in Figure 4-11 where the horizontal axis shows the first direction $[j]$ while the vertical axis shows the second direction $[k]$. The axes are labelled with the appropriate compass and Freeman chain code directions.

Second Direction [k]	First Direction [j]							
	N (0)	NE (1)	E (2)	SE (3)	S (4)	SW (5)	W (6)	NW (7)
N (0)								
NE (1)		**c**						
E (2)			**a**					
SE (3)				**A**	**B**			
S (4)		**d**	**b**	**C**	**D**			
SW (5)								
W (6)								
NW (7)								

Direction Histogram Bins

East to South Corner (A=0)

i	a max-value	b max-value	D max-value
1	5	0	0
2	4	1	0
3	3	1	1
4	2	1	2
5	**1**	**1**	**3**
6	0	1	4
7	0	0	5

North-East to South Corner (a=0)

i	c max-value	d max-value	D max-value
1	5	0	0
2	4	1	0
3	3	1	1
4	2	1	2
5	**1**	**1**	**3**
6	0	1	4
7	0	0	5

Direction Histogram Bin Contents

Figure 4-11: Contour Direction Histogram Bin Contents

Let (i) be the index of the point on the contour (Figure 4-12). Consecutive pairs of contour points are indexed from $(i-n)$ to $(i+n)$, where the value of n is specified at the start of the histogram collection. As the value of n increases the detail stored in the histogram will become coarser. Typically $n = 5$, but n may also be a function of the number of original points on the contour, or a set of values when a type of multi-resolution analysis is being performed. The element of the direction histogram $[j,k]$ is incremented for each pair of contour points, where j = direction at

$(i)+m$, and $k=$ direction at $(i+1)+m$ for $m=-n\ to+n$. The direction histogram therefore contains the curvature or gradient details over a section of the contour of size $(2n+1)$ points. Because the contour is a closed, wraparound at the start and finish of the stored contour points will not introduce any discontinuities at these extreme data points. *N* direction histograms are collected over the whole contour where *N* equals the number of sample points along the contour.

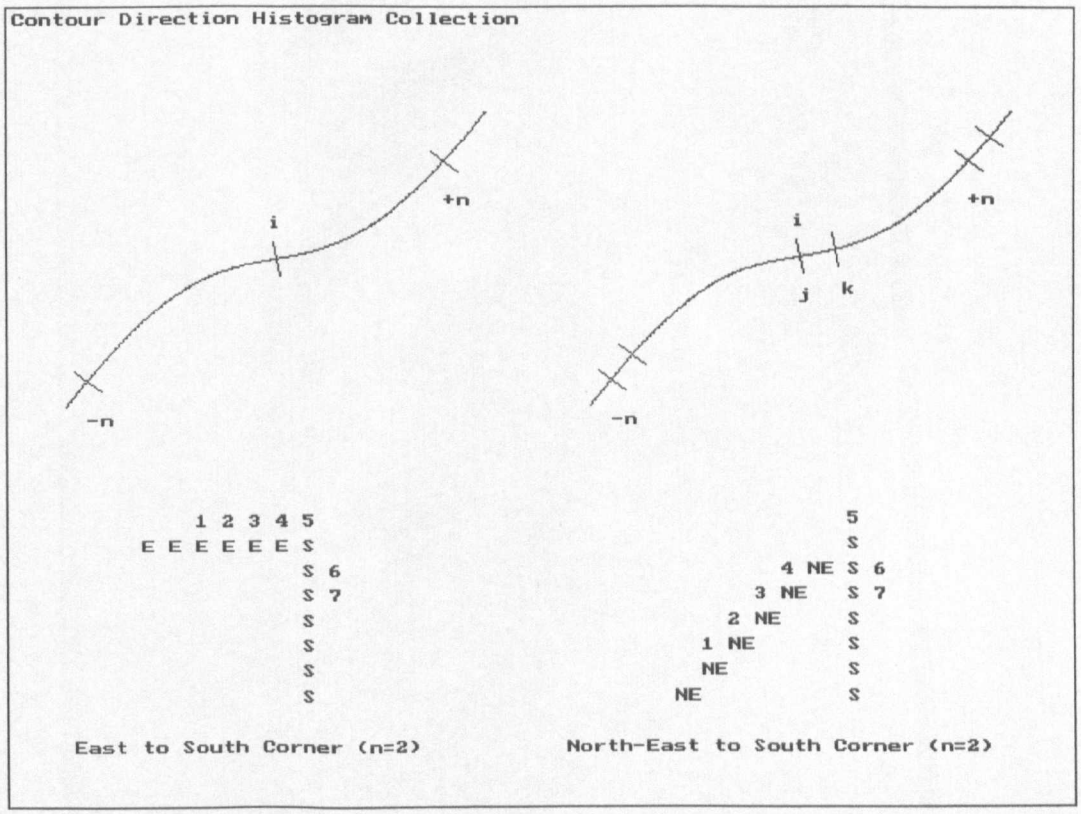

Figure 4-12: Contour Direction Histogram Collection

A PIP % calculation is performed that indicates how many times the change in direction at a point on the contour is contained within the main-diagonal of the direction histogram. Non-zero bins off the main-diagonal indicate discontinuities in the contour direction at a single point on the contour. Dividing this count by the number of points on the contour gives the fraction of the contour that changes smoothly at each point on the contour. Figure 4-13 shows the PIP % for an aircraft contour as the spatial frequency is varied. The PIP % is greater than 90% for spatial frequencies between 10 and 100 cycles per contour. Therefore approximately 10% of the contour points have discontinuities

associated with them between these spatial frequencies and this percentage will also be an estimate for the number of PIPs along the contour.

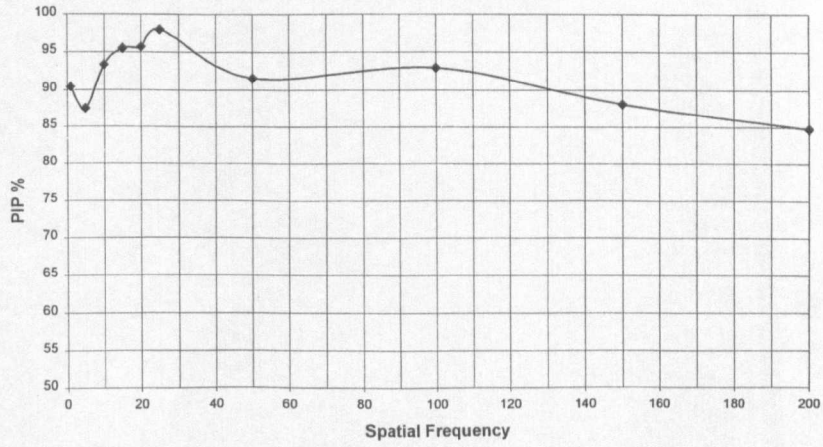

Figure 4-13: Aircraft Contour : PIP % versus Spatial Frequency

4.6.2 Principal Sub-Histograms

Eight sub-histograms are now defined along the main diagonal, as shown by the example **A**, **B**, **C**, and **D** (Figure 4-11). The sum of the histogram elements in each sub-histogram is calculated and the maximum value is noted together with its index in the direction histogram. If the example sub-histogram (**A**) were the maximum value, the data recorded would be [*max-index, max-value*], i.e. [3, *max-value*]. The values for *max-index* range from 0 to 7 and correspond to the histogram index for the bin designated as **A** in the example in Figure 4-11. The maximum possible value for *max-value* will be $(2n+1)$, when all the change-in-direction information is contained in one sub-histogram in the direction histogram. There will, therefore, be one sub-histogram for each point on the contour (a total of N sub-histograms), which may contain the maximum or principal change-in-direction of the contour over a region centred on each contour point. The number of points in this region is controlled by the value n.

The lower half of Figure 4-12 shows two examples of discontinuities on a contour, an East to South Corner and a North-East to South Corner. The lower half of Figure 4-11 shows the contents of the

direction histogram and the principal sub-histograms as the sample point traverses (clockwise) the discontinuity (corner) on the contour. The value of n equals 2, i.e. 5 points are sampled on the contour (± 2 points). Bin **a**, for the East to South Corner, initially contains the value 5. This value decreases around the corner to the value zero. Bin **b** contains the value 1 while the corner point is contained within the 5 samples. Bin **D** increases in value until the sample point has passed through the corner at which point it contains the value 5. Bins **a** and **D** are principal sub-histograms and indicate the principal directions of the contour over the sample length (5). A non-zero value in the bin **b** (off main-diagonal) indicates that a discontinuity is present on the contour. The row marked in bold, for $i = 5$, shows the point at which the discontinuity is identified. The maximum count in the principal sub-histogram changes from **a** to **D** at this point. A similar process occurs for the North-East to South Corner where the principal sub-histogram bins are **c** and **D**, and the off main-diagonal bin is **d**. The change in the principal sub-histogram index (from **a** to **D** or **c** to **D**) indicates the presence of a PIP on the contour, i.e. from 2 to 4 and 1 to 4 respectively.

Figure 4-10 shows the original shape of an aircraft contour (200 spatial frequencies), and Figure 4-14 shows the first normalised spatial frequency for this aircraft contour. Note that the original starting point (indicated by a cross) for the contour has been normalised to the position at the right hand end of the major axis for the ellipse.

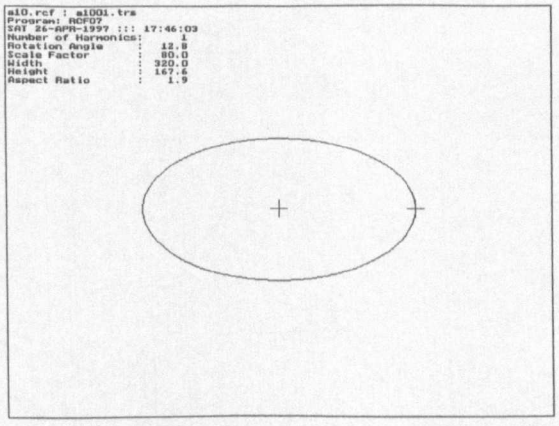

Figure 4-14: Aircraft Contour : 1 Spatial Frequency

Figure 4-15 shows an example set of PIPs for the first normalised spatial frequency and displays the index of the principal (maximum valued) sub-histogram as the contour is traversed in a clockwise direction. This index is traced in the lower half on the left of the figure, with the vertical axis showing the values 0, 3 and 7. These PIPs correspond to the positions on the contour, where the principal sub-histogram index changes. These points are plotted above the sub-histogram index display and identified by the crosses superimposed on the filtered contour trace.

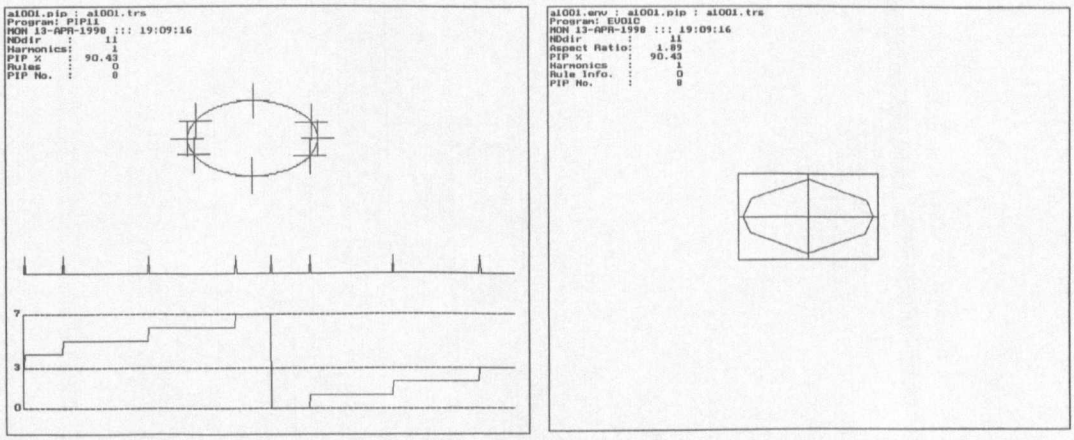

Figure 4-15: Aircraft Contour : 1 Spatial Frequency : Rule 0 : PIP - PIP Directions

4.6.3 Principal Sub-Histogram Rules

A number of rules can now be defined in order to identify the various PIPs along a contour. Two simple rules are investigated by the research work in this chapter that are able to provide information about a contour in a suitable form for input to a genetic algorithm. The rules are specified as follows:

1. Rule 0 measures the change in the index for the principal sub-histogram and, hence, measures the positions on the contour at which the index of the maximum valued sub-histogram changes. This rule captures most of the PIP information along a contour. This first level of PIP identification effectively obtains the coarsest positions for the PIP along the contour. Further rules are required to identify and extract the PIP between the above coarse PIP positions along the contour.

2. Rule 1 considers how the contents of **A**, **B**, **C** and **D** of the principal sub-histogram behave along the contour (Figure 4-11). A secondary or less coarse PIP is identified when the values (**A**+**C**) and (**B**+**D**), within the principal sub-histogram, change their values relative to each other. A secondary PIP is identified when (**A**+**C**) > (**B**+**D**) changes to the condition (**B**+**D**) ≥ (**A**+**C**). This rule (1) identifies the position along the contour when the predominant directions within the principal sub-histogram change relative to one another.

Other rules are possible and recommended further research is discussed in Chapter 9.

A display of the first spatial frequency for an aircraft contour, using rule 1, is shown inFigure 4-16. Note that the coarse PIP curve (Figure 4-15) has not changed, but extra PIPs are added to the contour, indicated by the extra crosses displayed on the contour, and by the extra points on the PIP plot above the main PIP information lower display. Note also that the PIP % does not change when applying rule 1. This is because rule 1 introduces extra PIP information that is calculated from changes within a principal sub-histogram, whereas the PIP % is calculated using the off main-diagonal direction histogram bins.

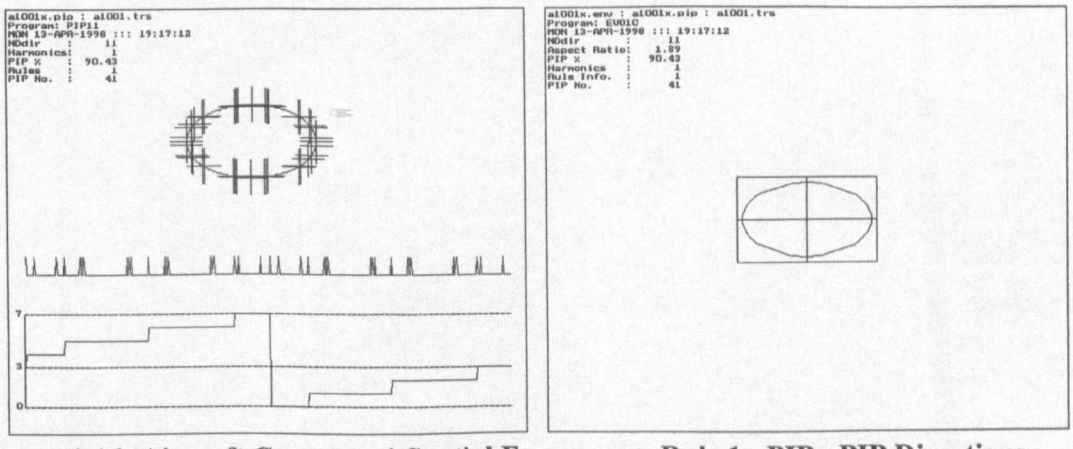

Figure 4-16: Aircraft Contour : 1 Spatial Frequency : Rule 1 : PIP - PIP Directions

This new method for finding the PIPs on a contour has been developed with processing times in mind. Matching polygons and cubic splines to the curvature of a contour appear to be unnecessarily

complex methods for finding the PIP information on a contour. Approximate positions and directions for the PIPs may be all that are usually required for a recognition process, and where possible a multi-scale and a coarse to fine contour sampling should be made available when using any particular method. The positions of the PIPs on a contour do not have to be identified exactly because further processing only resolves the PIP directions into the 8 Freeman chain code directions.

4.6.4 Perceptually Important Point Characteristics

This research has also investigated two characteristics of this new method for finding PIPs along a contour. These two characteristics provide a check on the new method and these characteristics should to be present for a PIP calculation technique to be considered suitable for use with the genetic algorithm developed in Chapter 5. These two characteristics are listed as follows:

1. The first characteristic concerns the ability of the new method to identify the PIP as the contour is reconstructed with an increasing number of spatial frequencies (multi-scale).
2. The second characteristic concerns the ability of the new PIP identification method to identify the appropriate PIP information correctly as the sampling length increases (coarse/fine).

Various experiments were performed to confirm that the new technique for PIP calculation had these two characteristics and the results are summarised in Figure 4-14 to Figure 4-71.

Figure 4-14 to Figure 4-43 show the PIP positions, on an aircraft contour, for a sample of spatial frequencies varying from 1 to 200 cycles/contour. Examples for the rules 0 and 1 are included in the figures. The original contour is shown in Figure 4-10, and the rest of the figures are then in groups of five showing the normalised reconstructed contour with the PIP positions for rules 0 and 1. Accompanying each PIP display is the corresponding processed PIP information, which has the direction for each PIP resolved to the 8 Freeman chain code directions. Note that rule 1 identifies the PIPs on the first spatial frequency ellipse more accurately than rule 0.

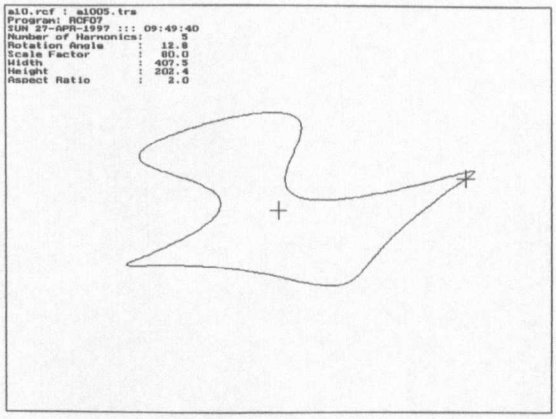

Figure 4-17: Aircraft Contour : 5 Spatial Frequencies

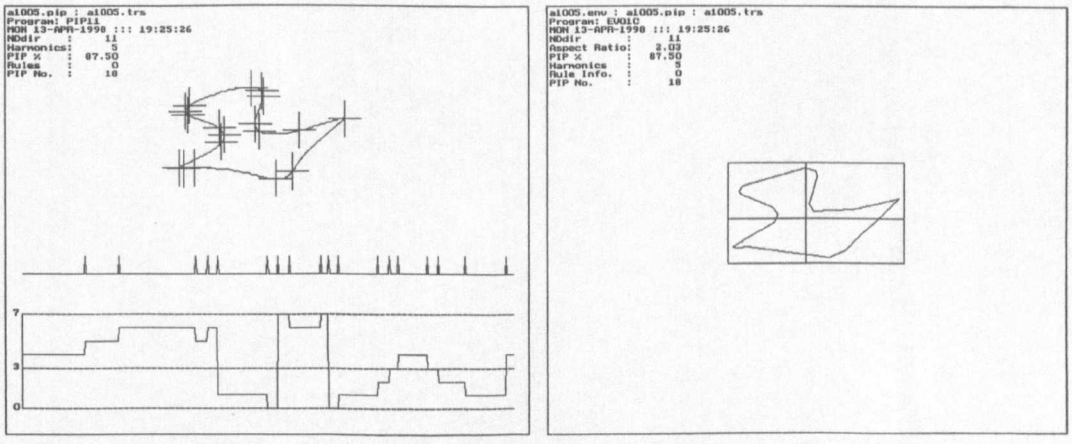

Figure 4-18: Aircraft Contour : 5 Spatial Frequencies : Rule 0 : PIP - PIP Directions

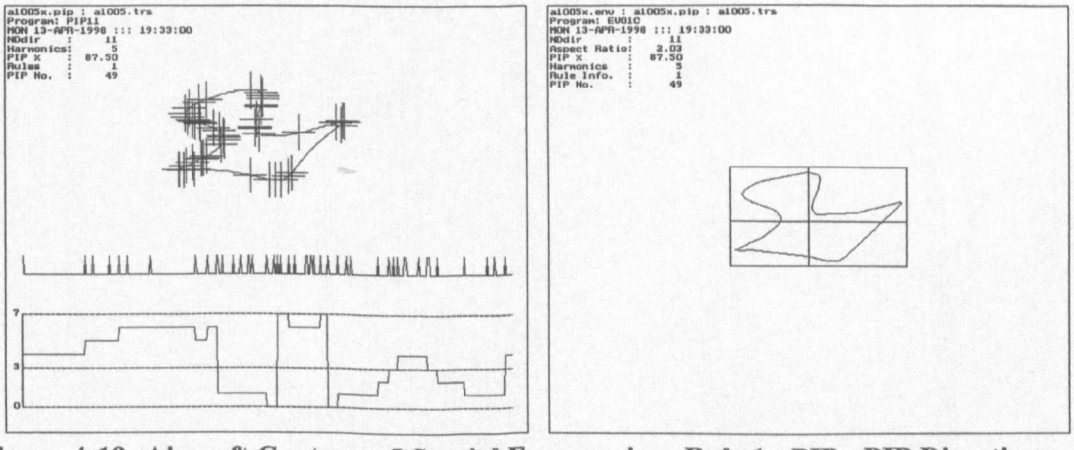

Figure 4-19: Aircraft Contour : 5 Spatial Frequencies : Rule 1 : PIP - PIP Directions

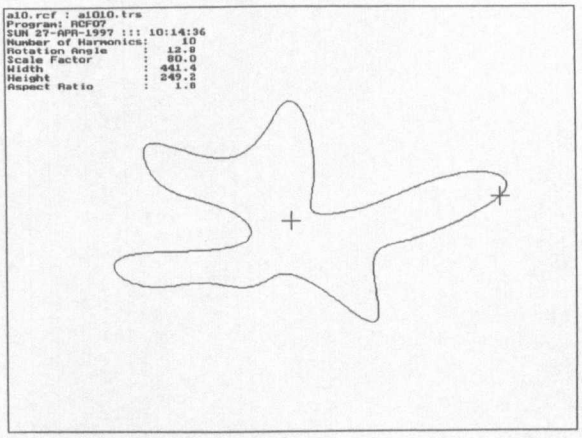

Figure 4-20: Aircraft Contour : 10 Spatial Frequencies

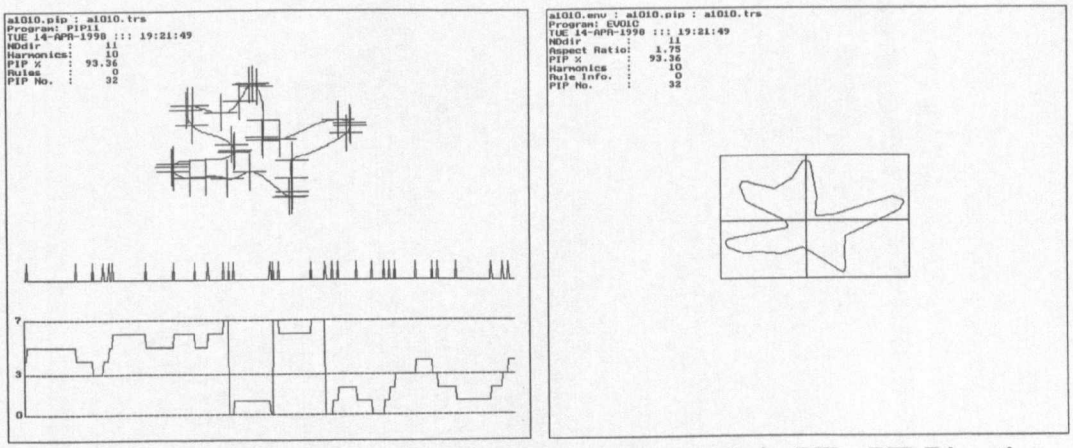

Figure 4-21: Aircraft Contour : 10 Spatial Frequencies : Rule 0 : PIP - PIP Directions

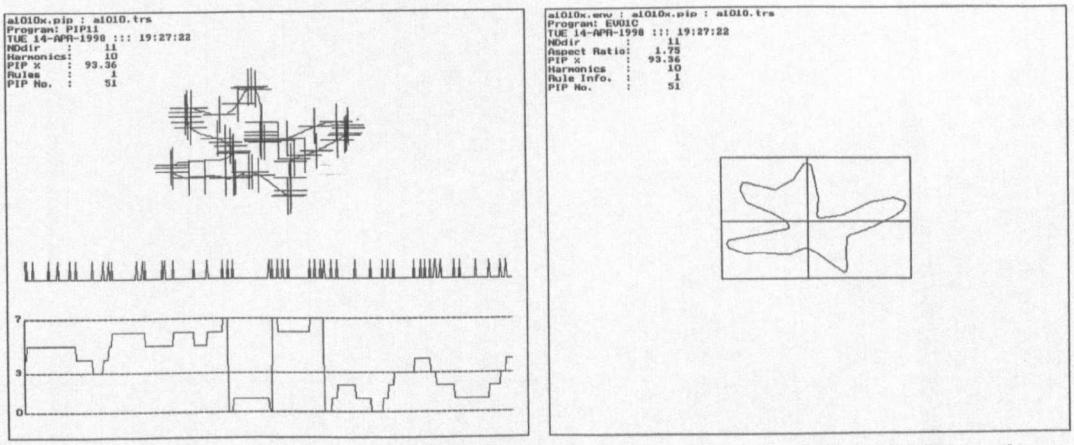

Figure 4-22: Aircraft Contour : 10 Spatial Frequencies : Rule 1 : PIP - PIP Directions

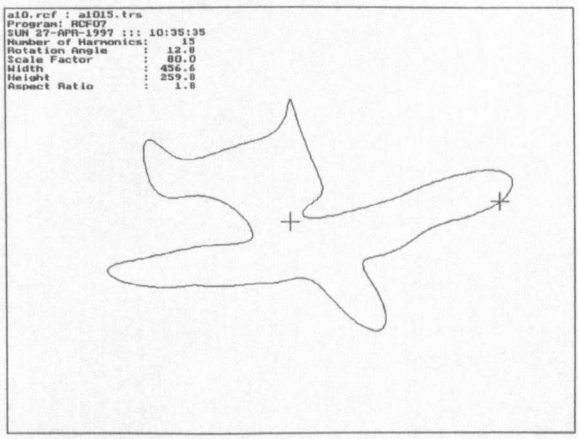

Figure 4-23: Aircraft Contour : 15 Spatial Frequencies

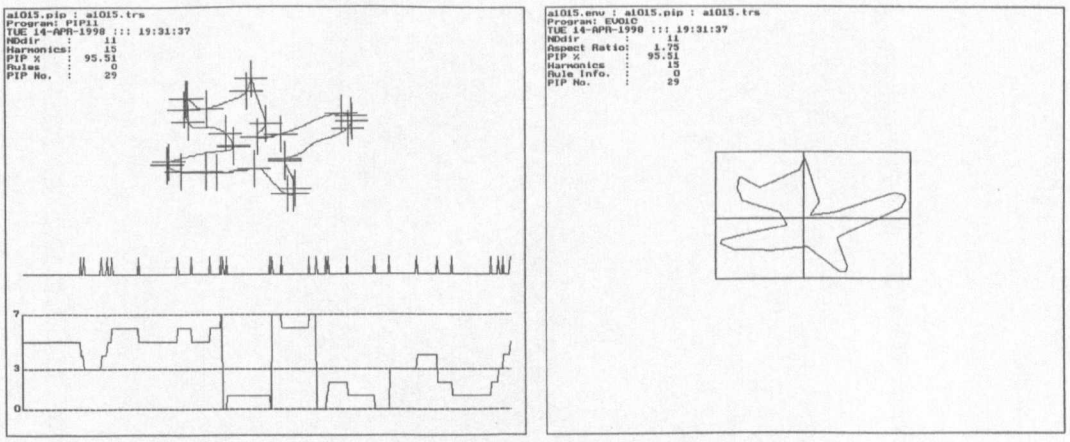

Figure 4-24: Aircraft Contour : 15 Spatial Frequencies : Rule 0 : PIP - PIP Directions

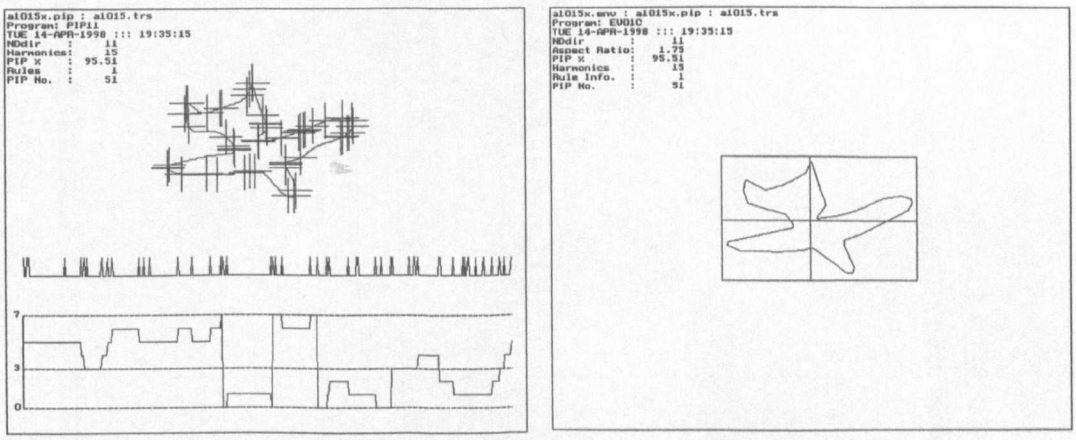

Figure 4-25: Aircraft Contour : 15 Spatial Frequencies : Rule 1 : PIP - PIP Directions

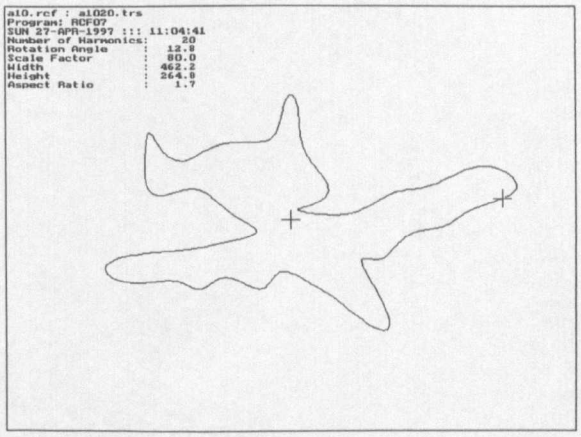

Figure 4-26: Aircraft Contour : 20 Spatial Frequencies

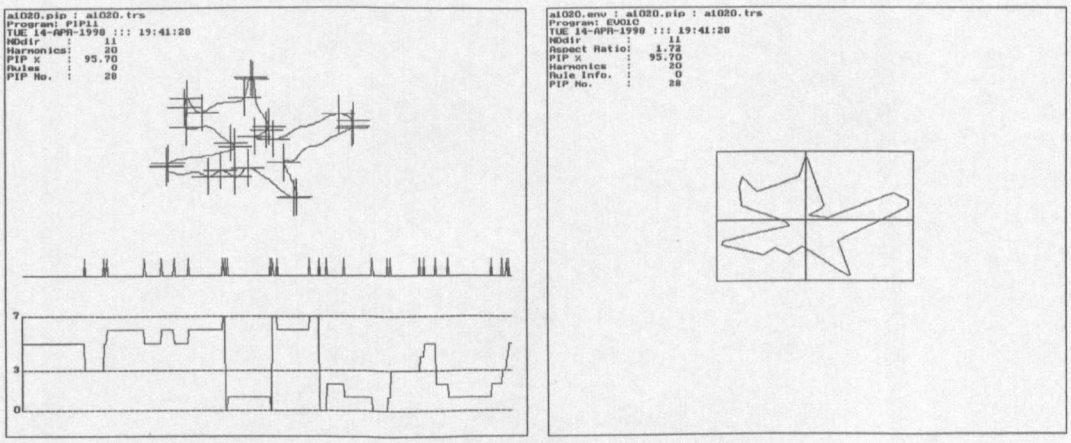

Figure 4-27: Aircraft Contour : 20 Spatial Frequencies : Rule 0 : PIP - PIP Directions

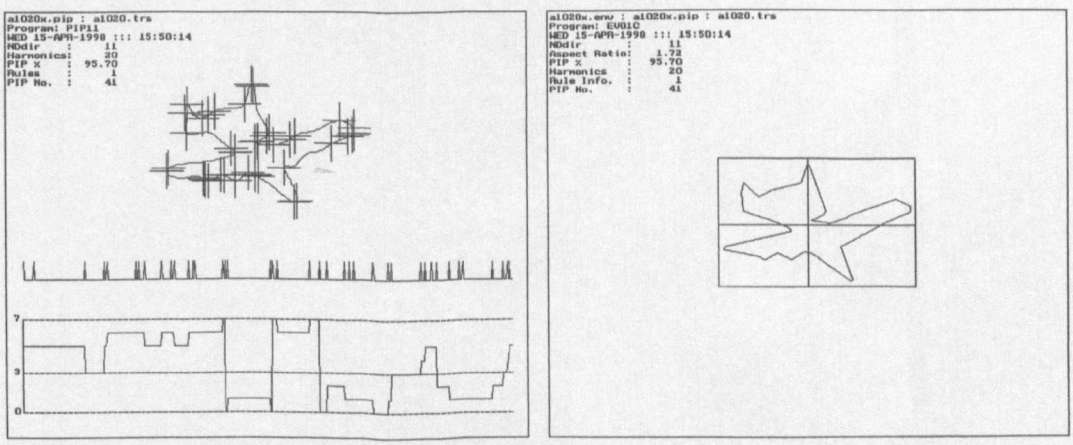

Figure 4-28: Aircraft Contour : 20 Spatial Frequencies : Rule 1 : PIP - PIP Directions

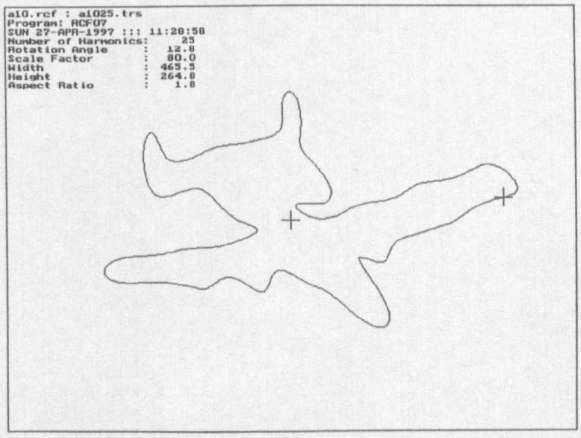

Figure 4-29: Aircraft Contour : 25 Spatial Frequencies

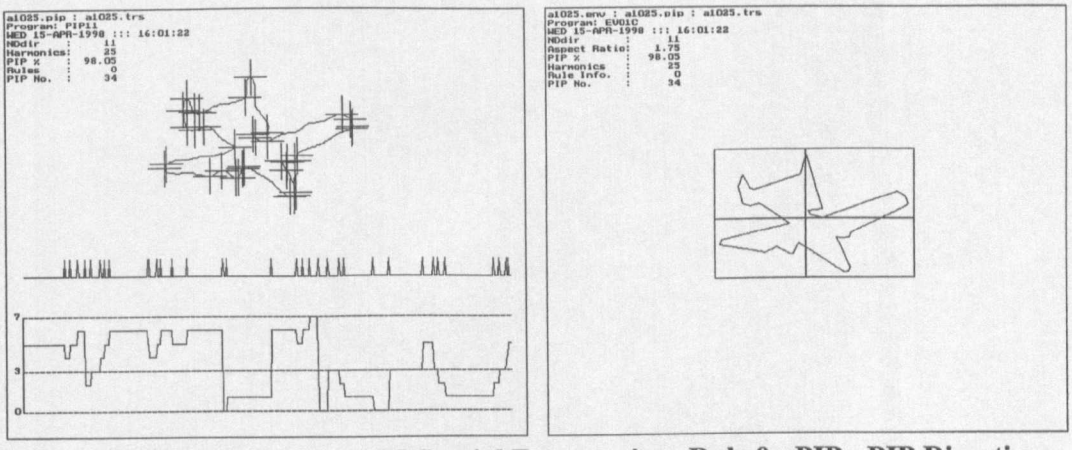

Figure 4-30: Aircraft Contour : 25 Spatial Frequencies : Rule 0 : PIP - PIP Directions

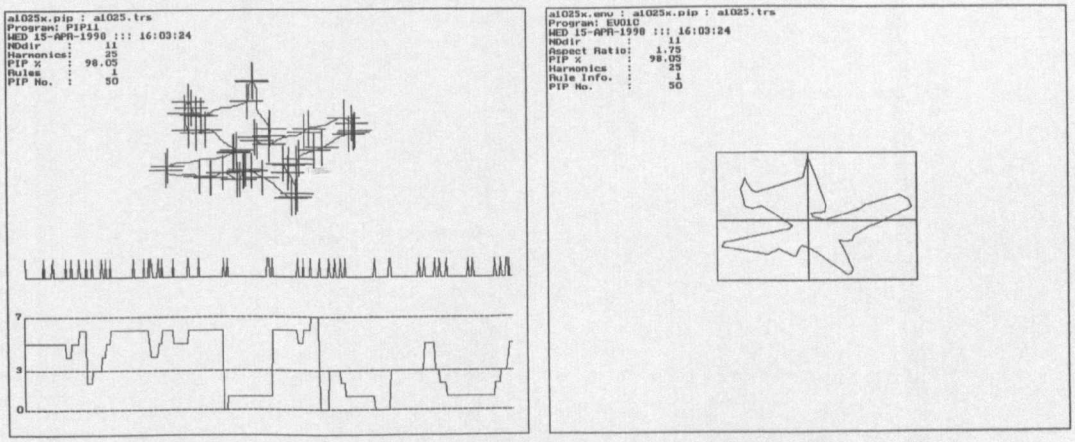

Figure 4-31: Aircraft Contour : 25 Spatial Frequencies : Rule 1 : PIP - PIP Directions

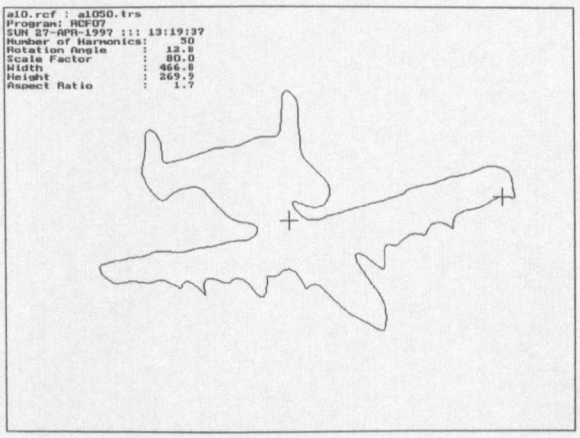

Figure 4-32: Aircraft Contour : 50 Spatial Frequencies

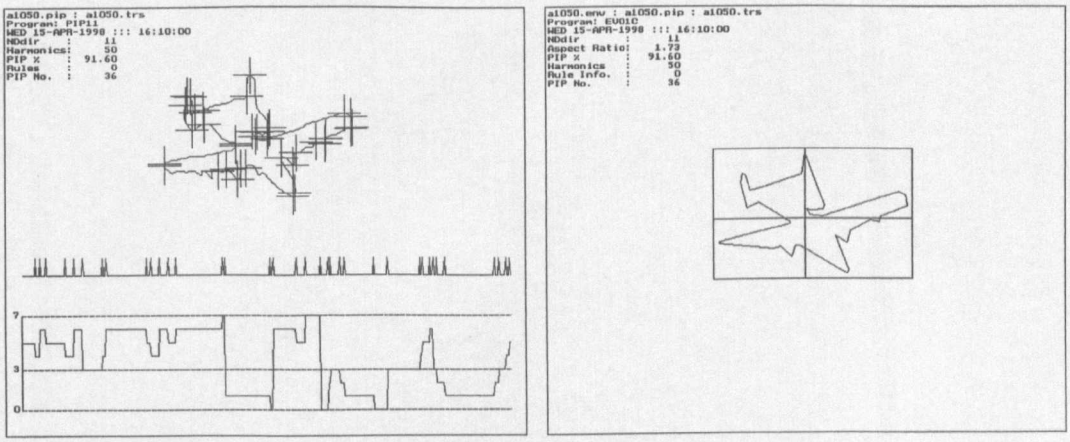

Figure 4-33: Aircraft Contour : 50 Spatial Frequencies : Rule 0 : PIP - PIP Directions

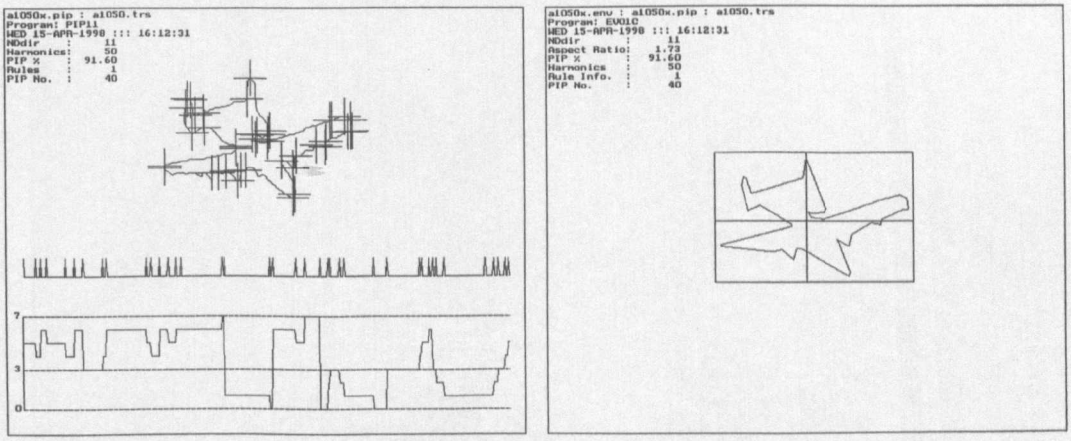

Figure 4-34: Aircraft Contour : 50 Spatial Frequencies : Rule 1 : PIP - PIP Directions

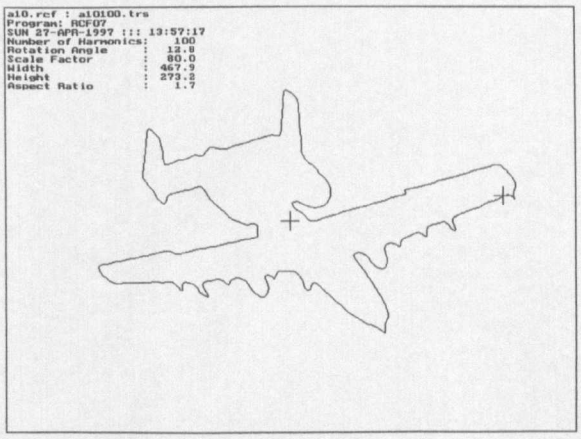

Figure 4-35: Aircraft Contour : 100 Spatial Frequencies

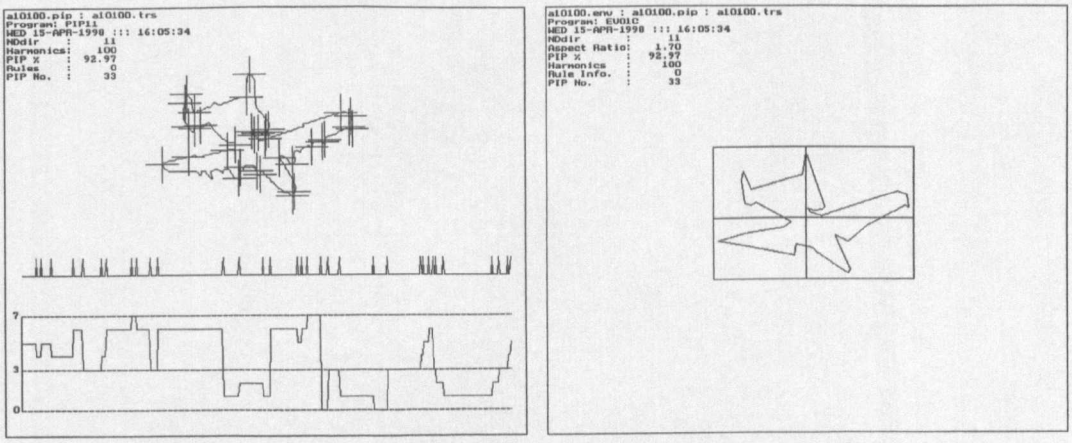

Figure 4-36: Aircraft Contour : 100 Spatial Frequencies : Rule 0 : PIP - PIP Directions

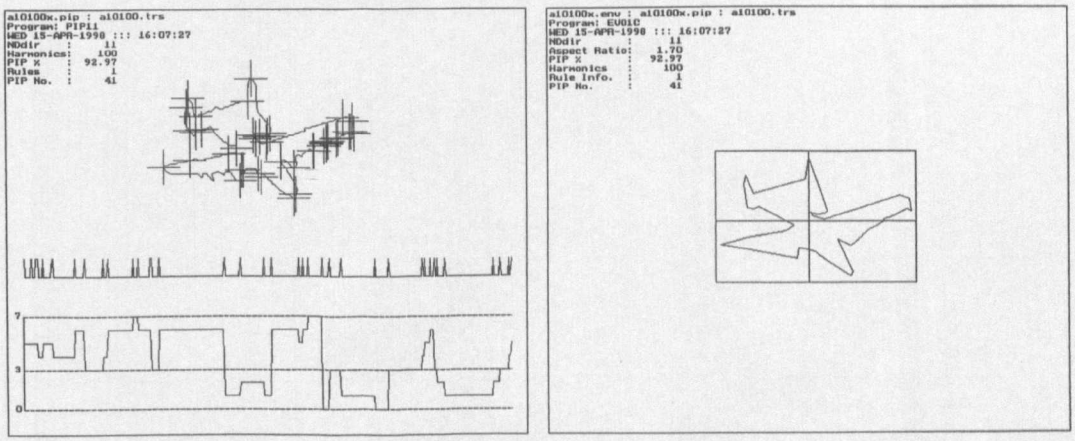

Figure 4-37: Aircraft Contour : 100 Spatial Frequencies : Rule 1 : PIP - PIP Directions

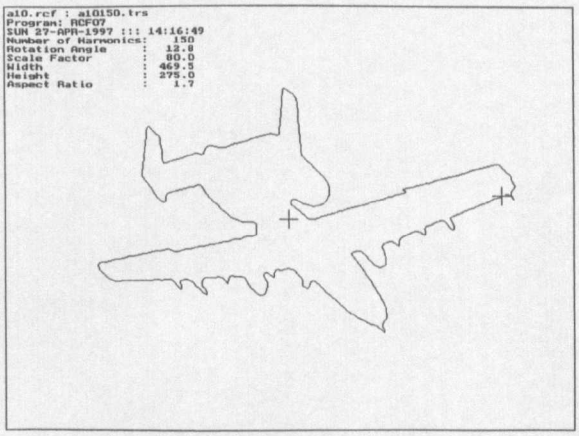

Figure 4-38: Aircraft Contour : 150 Spatial Frequencies

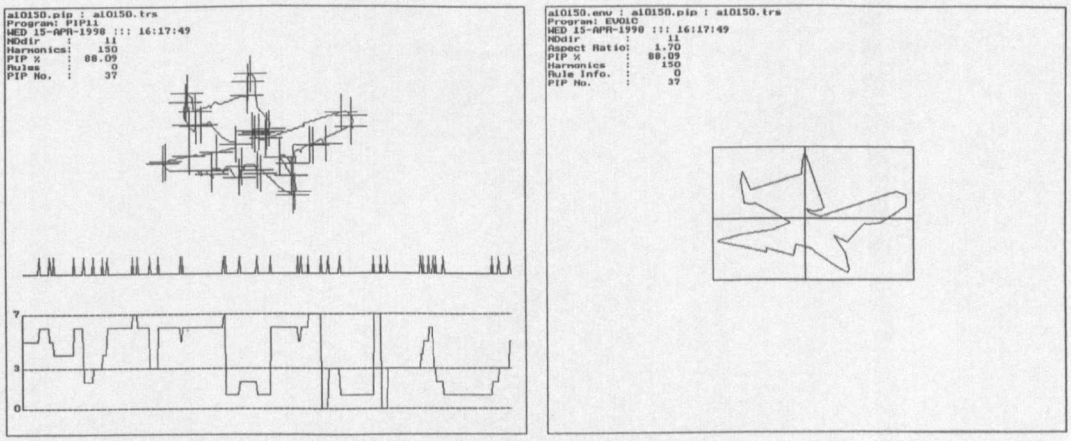

Figure 4-39: Aircraft Contour : 150 Spatial Frequencies : Rule 0 : PIP - PIP Directions

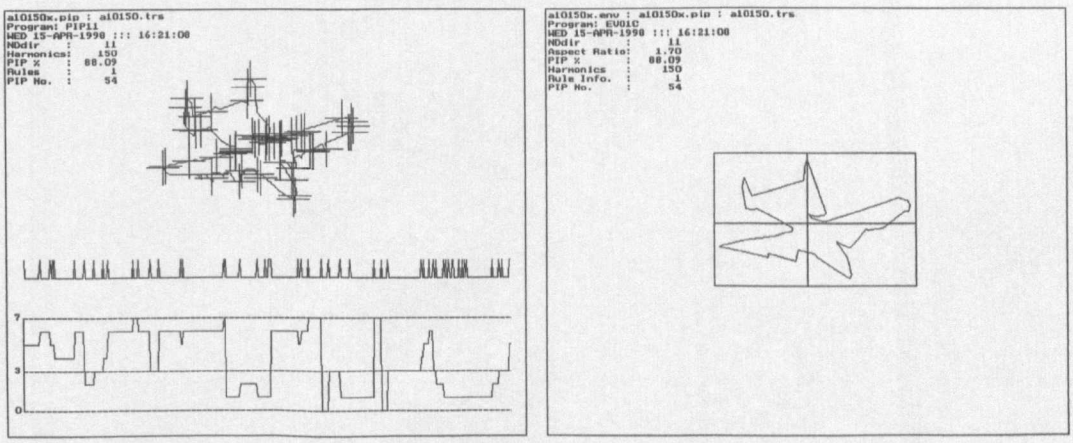

Figure 4-40: Aircraft Contour : 150 Spatial Frequencies : Rule 1 : PIP - PIP Directions

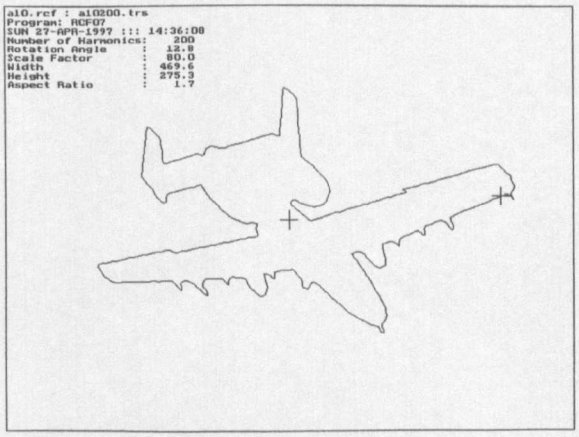

Figure 4-41: Aircraft Contour : 200 Spatial Frequencies

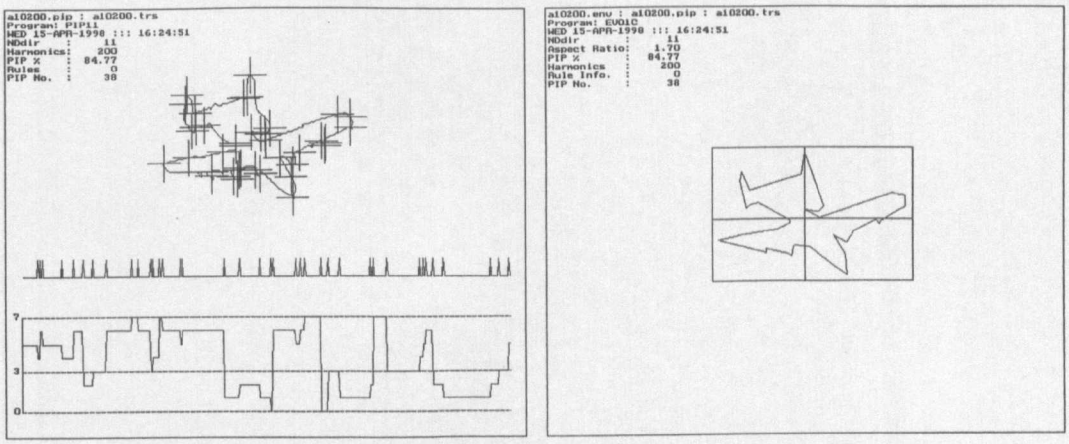

Figure 4-42: Aircraft Contour : 200 Spatial Frequencies : Rule 0 : PIP - PIP Directions

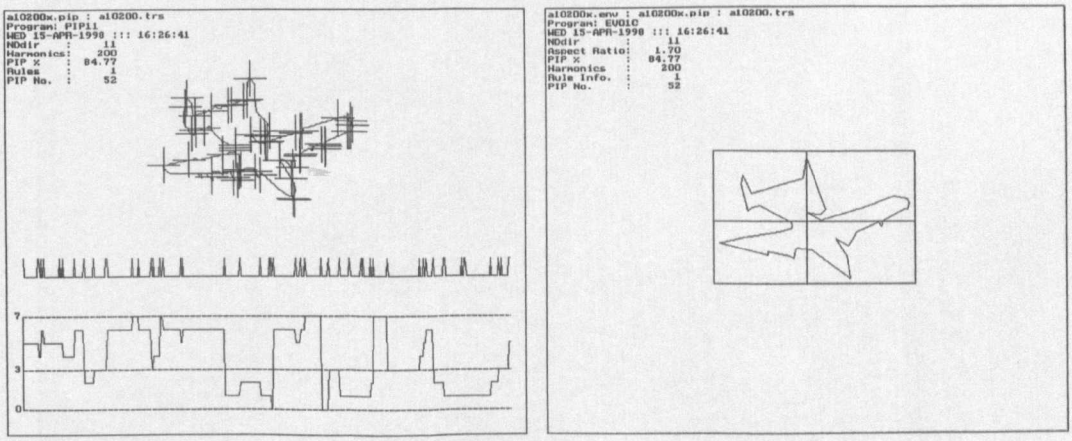

Figure 4-43: Aircraft Contour : 200 Spatial Frequencies : Rule 1 : PIP - PIP Directions

The underlying shape of the maximum valued sub-histogram index curve (at the lower half of the PIP display, vertical axis labelled 0, 3 and 7) remains basically the same for values greater than 25 cycles/contour for the spatial frequencies of the reconstructed contour. Rule 1 appears, in general, to

give a slightly better approximation to the contour. Note that this new method of PIP identification finds the PIPs on a contour but may not necessarily match the PIP directions exactly along the contour.

Two measurements can be made while collecting the PIP information along a contour. These measurements are listed as follows:

1. The percentage of the contour covered between the PIP.
2. The number of PIPs measured on the contour.

Experimental results using these measurements are discussed in the following paragraphs.

The experimental results for the aircraft contour are summarised in Figure 4-13 and Figure 4-44 to Figure 4-45. The first 50 spatial frequencies calculate the PIP % to be > 90%, which reduces to approximately 85% at a spatial frequency of 200 cycles/contour. The number of PIPs (Rule 0, Figure 4-44) increases up to the 25 cycles/contour and then varies in an approximate band between 30 and 40 PIPs. Hence reconstructing the contour with spatial frequencies about 25 cycles/contour provides a reasonable selection of PIPs. The number of PIPs for Rule 1, Figure 4-45, varies in a similar manner between 40 and 55 PIPs.

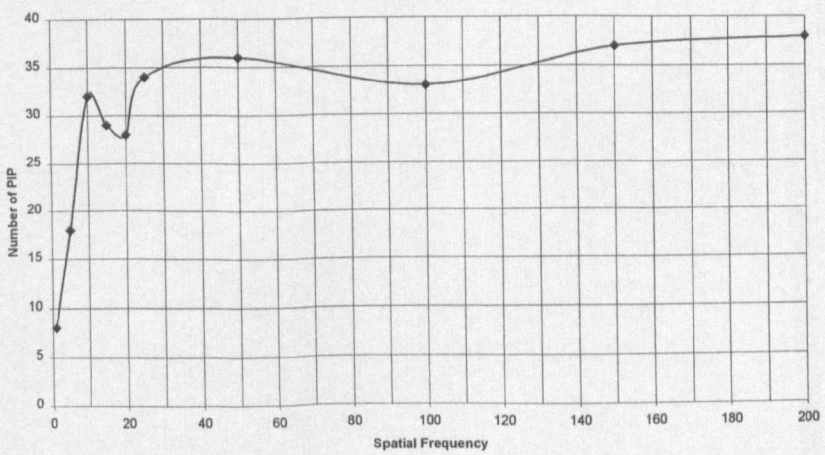

Figure 4-44: Aircraft Contour : Number of PIP versus Spatial Frequency : Rule 0

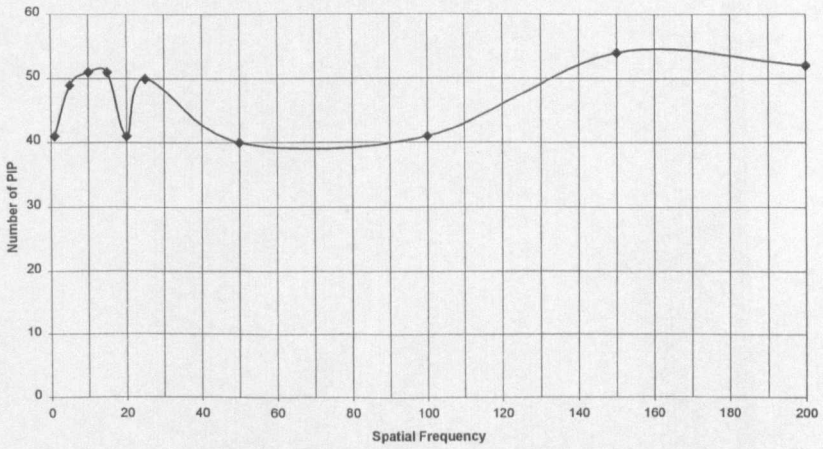

Figure 4-45: Aircraft Contour : Number of PIP versus Spatial Frequency : Rule 1

Figure 4-46 to Figure 4-54 show, for the aircraft contour, the PIP information and the PIP directions for increasing length of the sampling points, i.e. increasing value for n. Rule 1 was used in each case. The number of PIPs decreases as the length of the sample points increases. The 'best' set of PIPs, from visual inspection, appears for sampling lengths from ±2 to ±5 contour points. The results are summarised in Figure 4-55 to Figure 4-58. The PIP % (Figure 4-55) is > 98% for sample points ≤ ±5 (i.e. 11 sample points) along the contour. The number of PIPs reduces as the number of sample points increases (Figure 4-57) and at ±5 sample points gives approximately 50 PIP identifications.

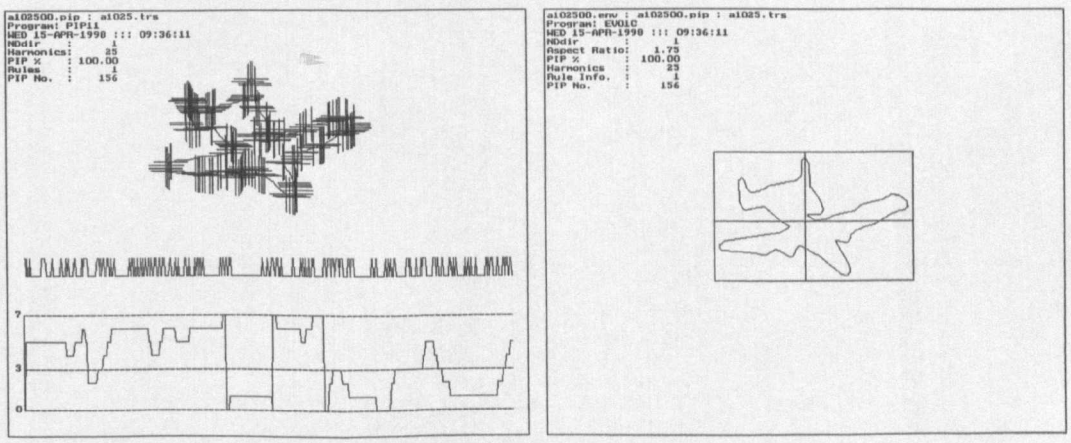

Figure 4-46: Aircraft Contour : 1 Sample Point : PIP - PIP Directions

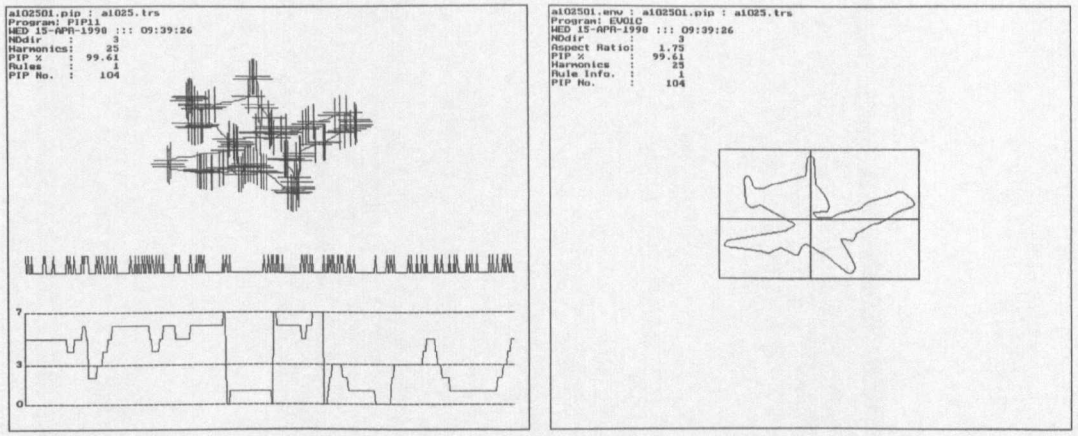

Figure 4-47: Aircraft Contour : 3 Sample Points : PIP - PIP Directions

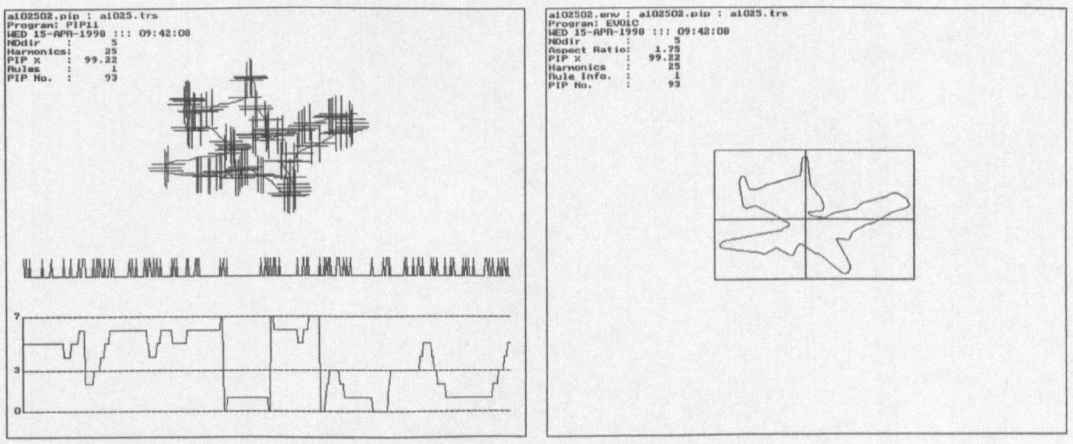

Figure 4-48: Aircraft Contour : 5 Sample Points : PIP - PIP Directions

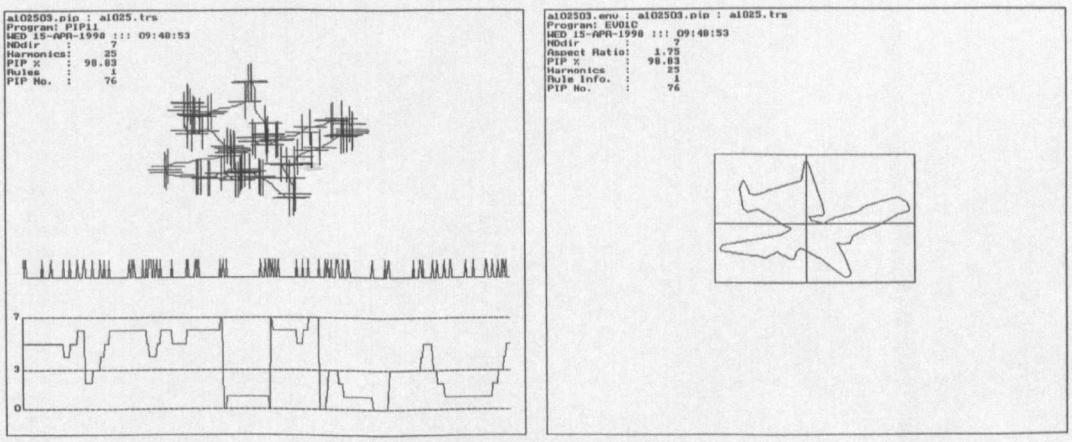

Figure 4-49: Aircraft Contour : 7 Sample Points : PIP - PIP Directions

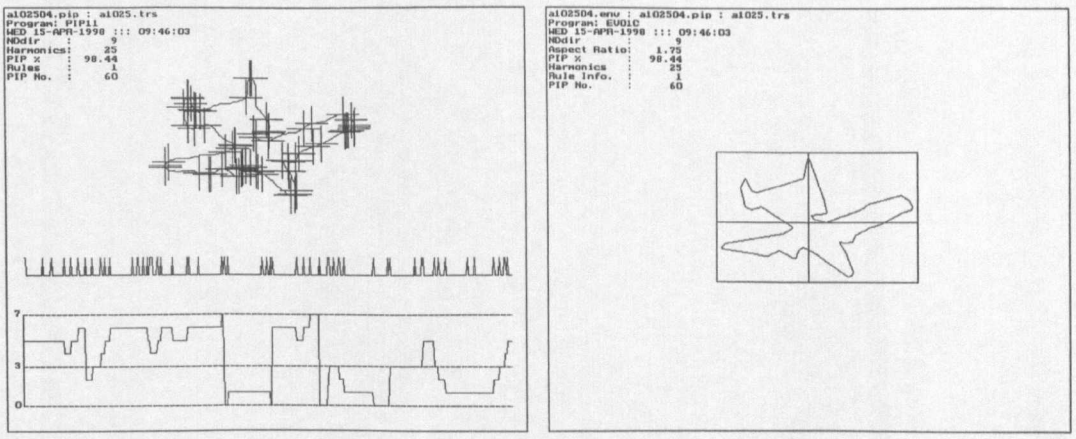

Figure 4-50: Aircraft Contour : 9 Sample Points : PIP - PIP Directions

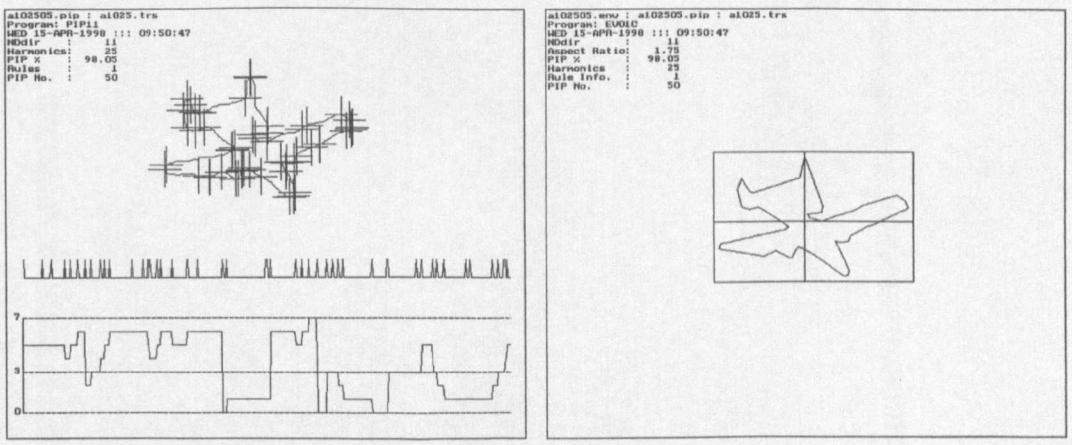

Figure 4-51: Aircraft Contour : 11 Sample Points : PIP - PIP Directions

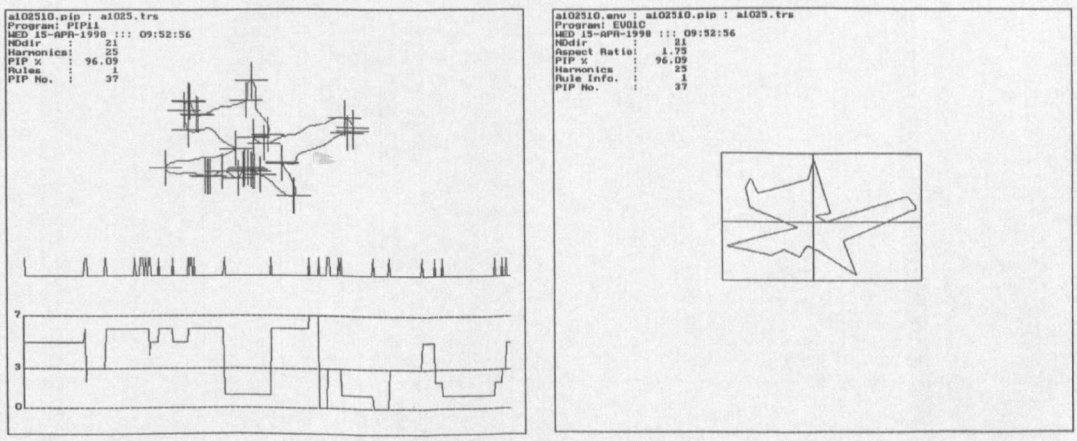

Figure 4-52: Aircraft Contour : 21 Sample Points : PIP - PIP Directions

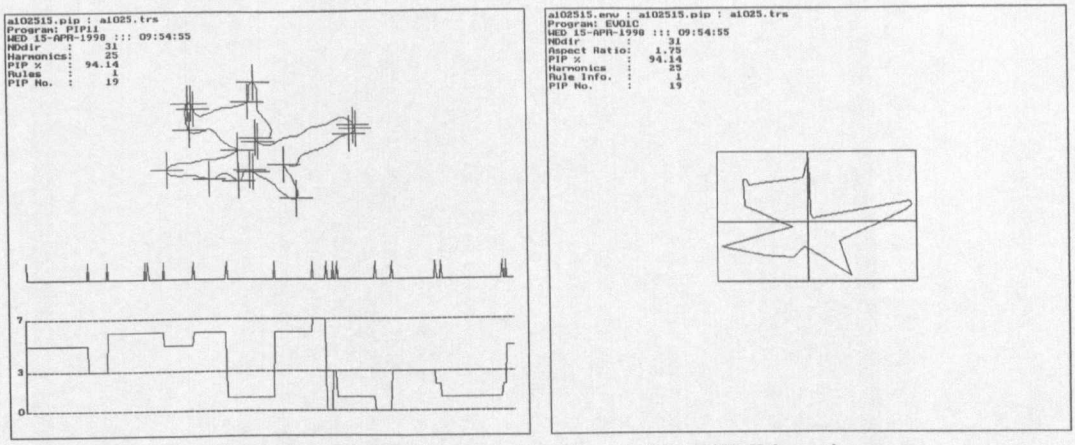

Figure 4-53: Aircraft Contour : 31 Sample Points : PIP - PIP Directions

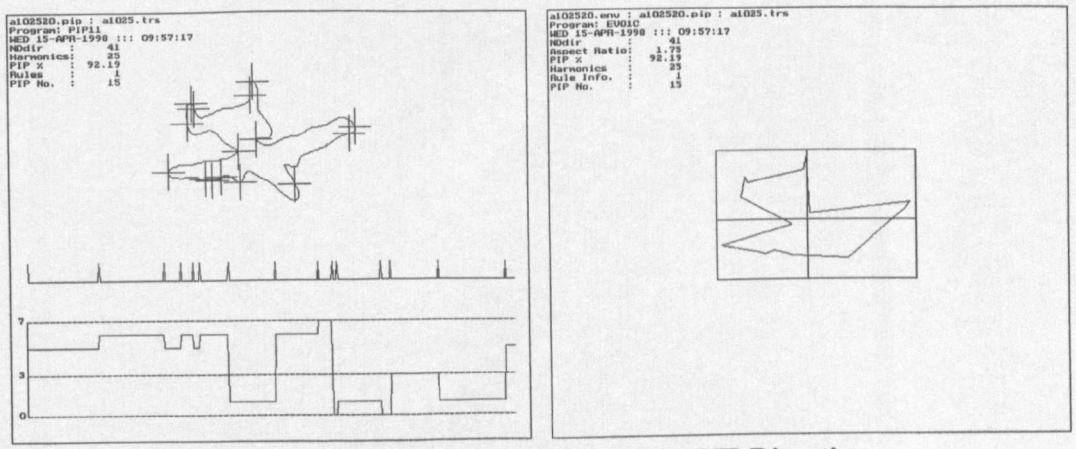

Figure 4-54: Aircraft Contour : 41 Sample Points : PIP - PIP Directions

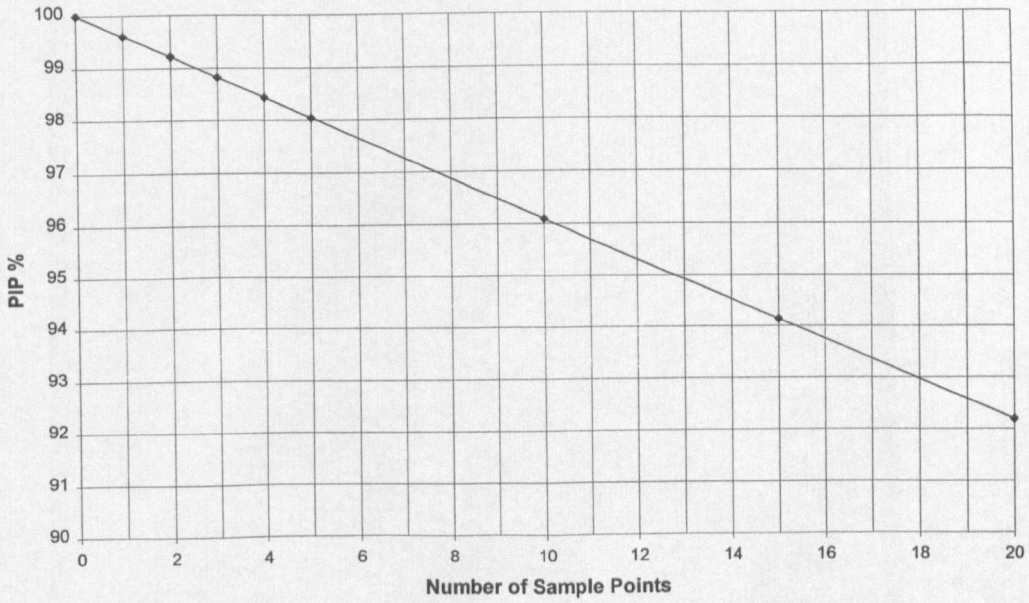

Figure 4-55: Aircraft Contour : PIP % versus Number of Sample Points

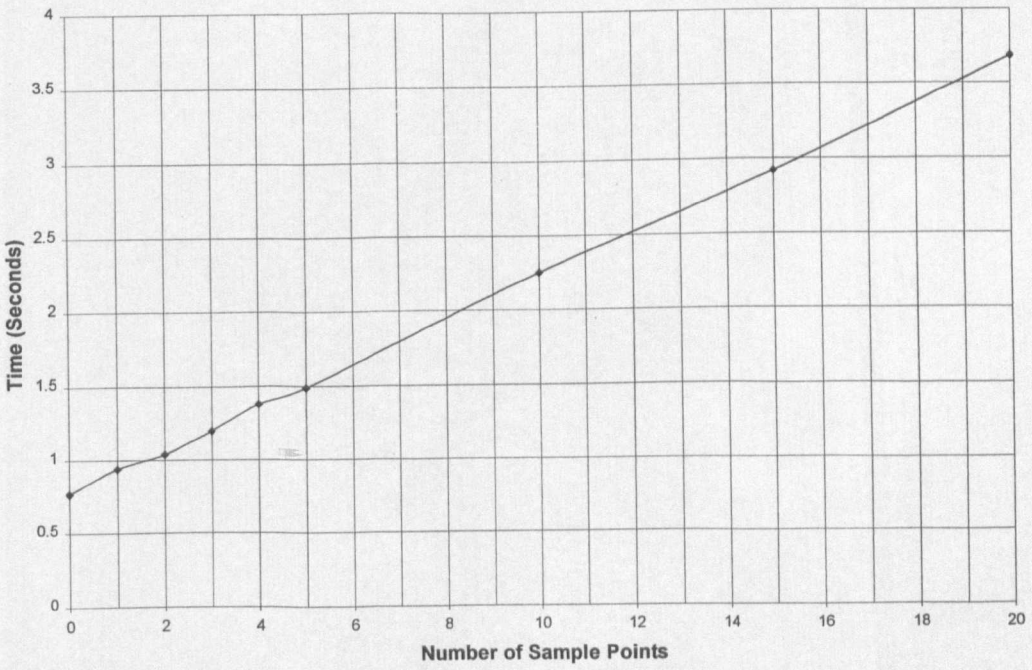

Figure 4-56: Aircraft Contour : PIP Time versus Number of Sample Points

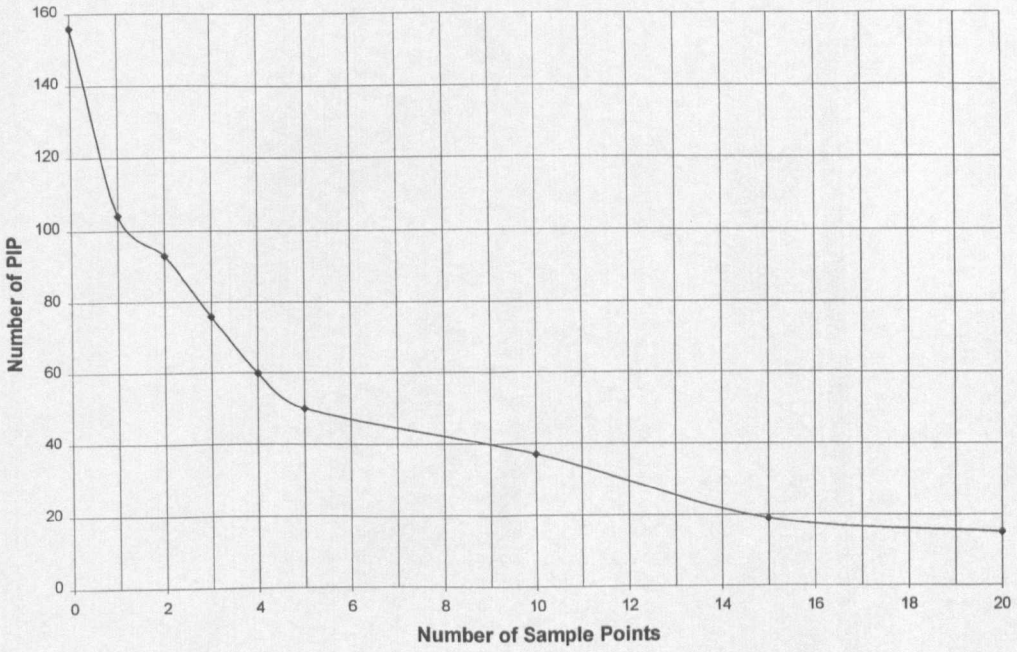

Figure 4-57: Aircraft Contour : Number of PIP versus Number of Sample Points

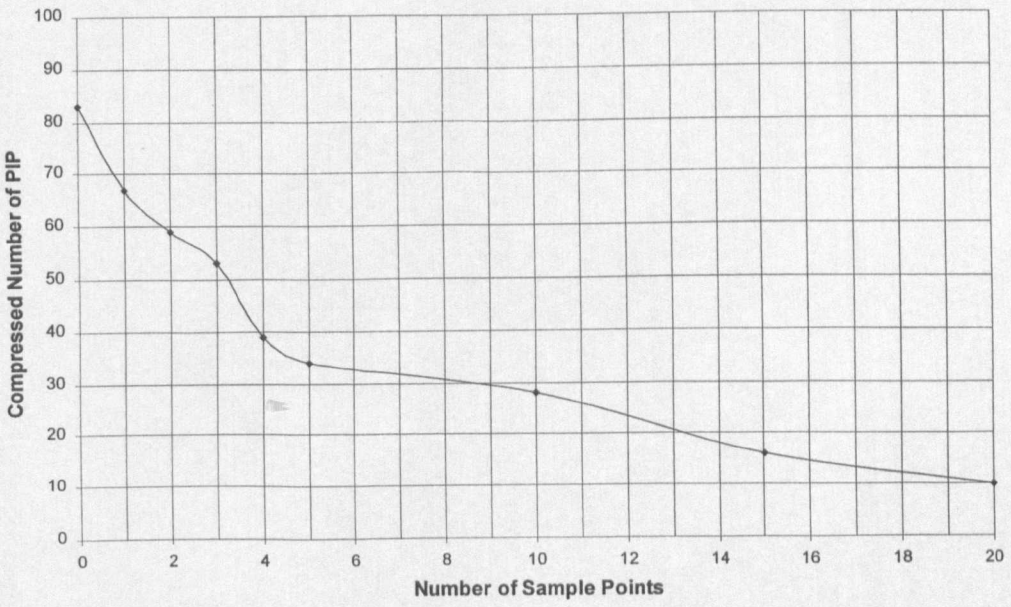

Figure 4-58: Aircraft Contour : Number of PIP (Compressed) versus Number of Sample Points

Compressed PIP data combines together line-segments with the same directions, hence effectively reducing the number of PIPs on the contour. Figure 4-58 shows the number of PIPs for compressed PIP data. Figure 4-56 shows the time taken to collect the PIP information as the number of sampling points increases.

Figure 4-59 to Figure 4-67 show similar results for the contour of the digit two. For the sample length of ±5 contour points the PIP % is > 85% (Figure 4-68) and the number of PIPs decrease to the value < 30 (Figure 4-70) at ±5 sample points along the contour. The number of PIPs for the compressed data reduces to less than 30 over the same number of sample points (Figure 4-71). The time to collect the PIP data (Figure 4-69) shows the same linear curve as that for the aircraft contour (Figure 4-56) and the time taken is the same for both contours. The same time is achieved by normalising all contours to the same number of points (512).

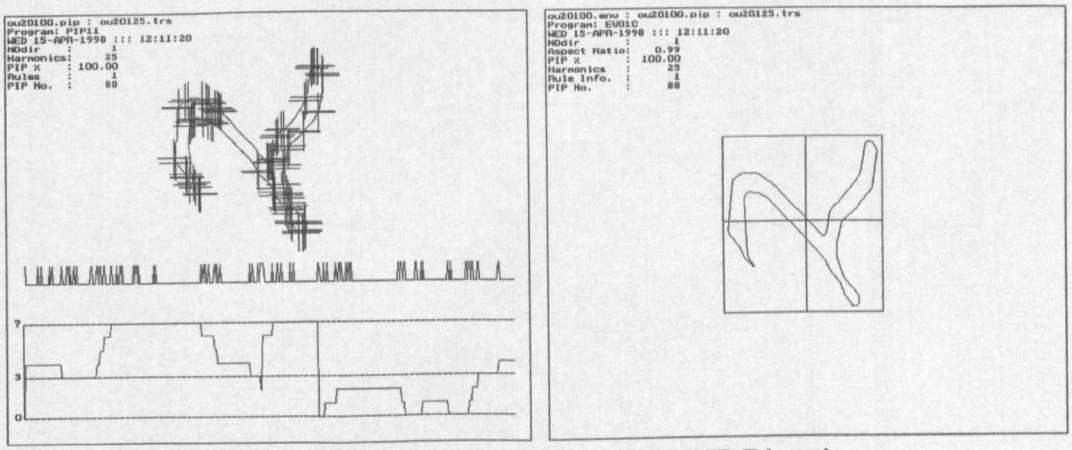

Figure 4-59: Digit Two Contour : 1 Sample Point : PIP - PIP Directions

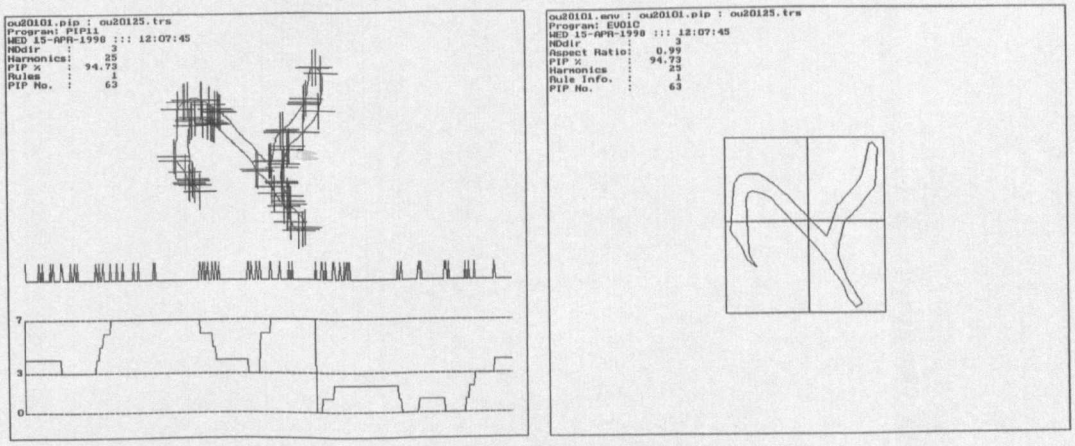

Figure 4-60: Digit Two Contour : 3 Sample Points : PIP - PIP Directions

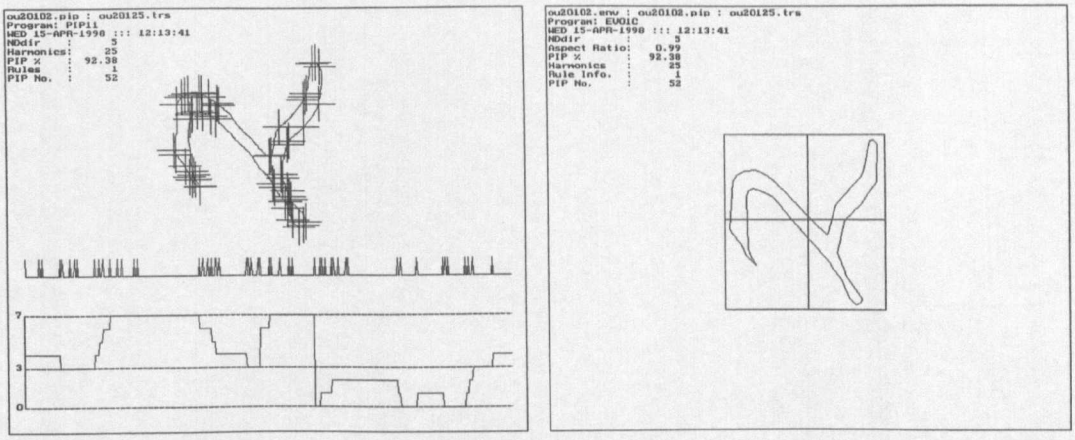

Figure 4-61: Digit Two Contour : 5 Sample Points : PIP - PIP Directions

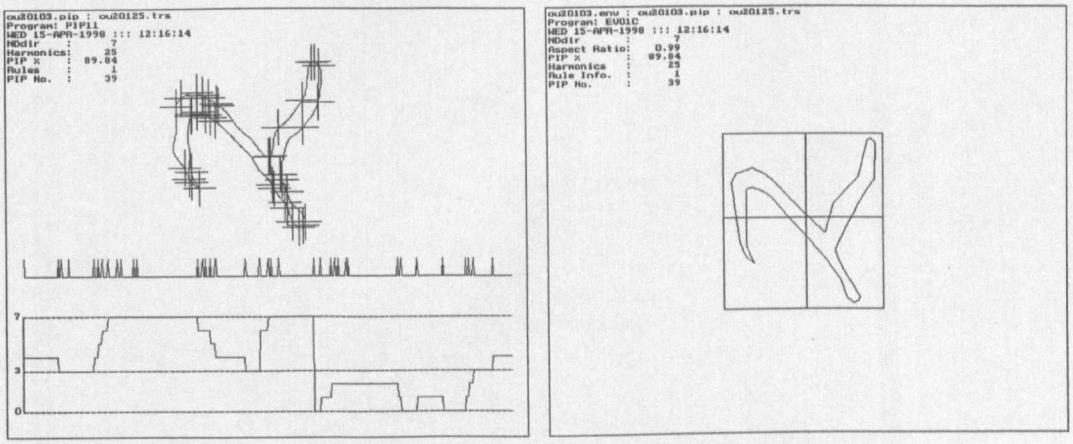

Figure 4-62: Digit Two Contour : 7 Sample Points : PIP - PIP Directions

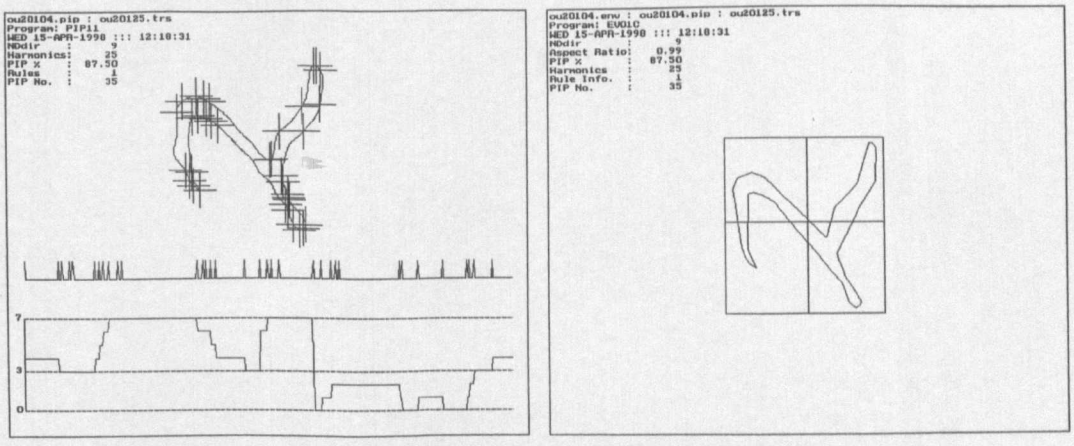

Figure 4-63: Digit Two Contour : 9 Sample Points : PIP - PIP Directions

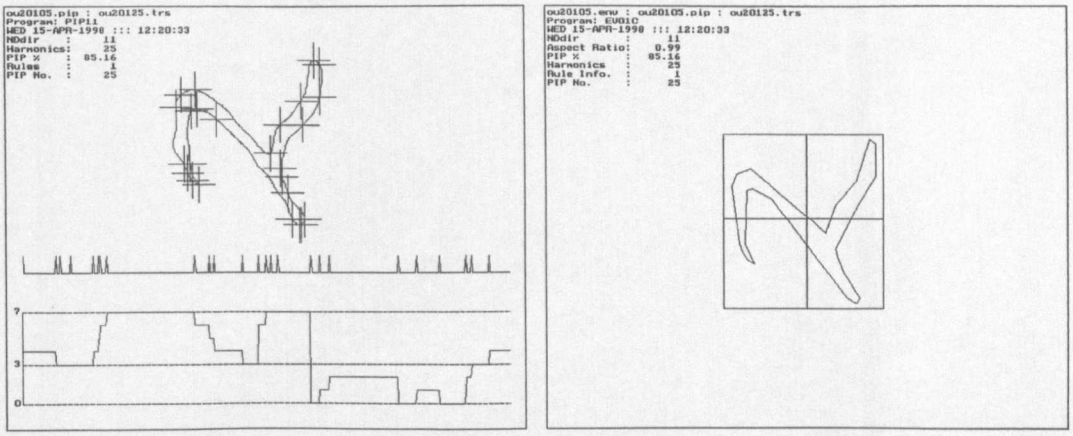

Figure 4-64: Digit Two Contour : 11 Sample Points : PIP - PIP Directions

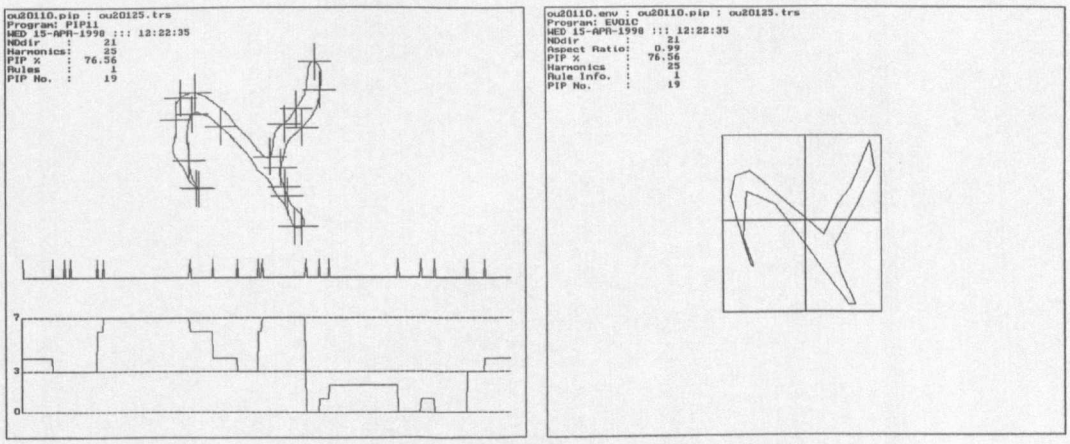

Figure 4-65: Digit Two Contour : 21 Sample Points : PIP - PIP Directions

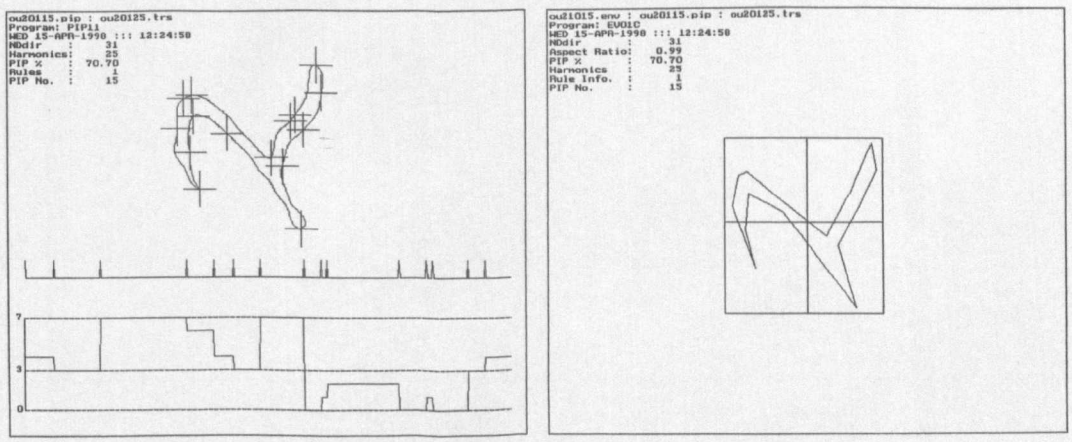

Figure 4-66: Digit Two Contour : 31 Sample Points : PIP - PIP Directions

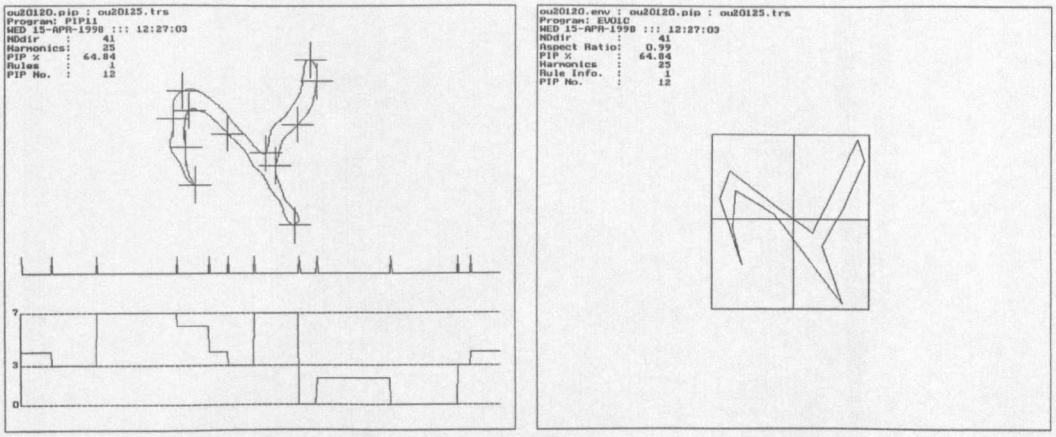

Figure 4-67: Digit Two Contour : 41 Sample Points : PIP - PIP Directions

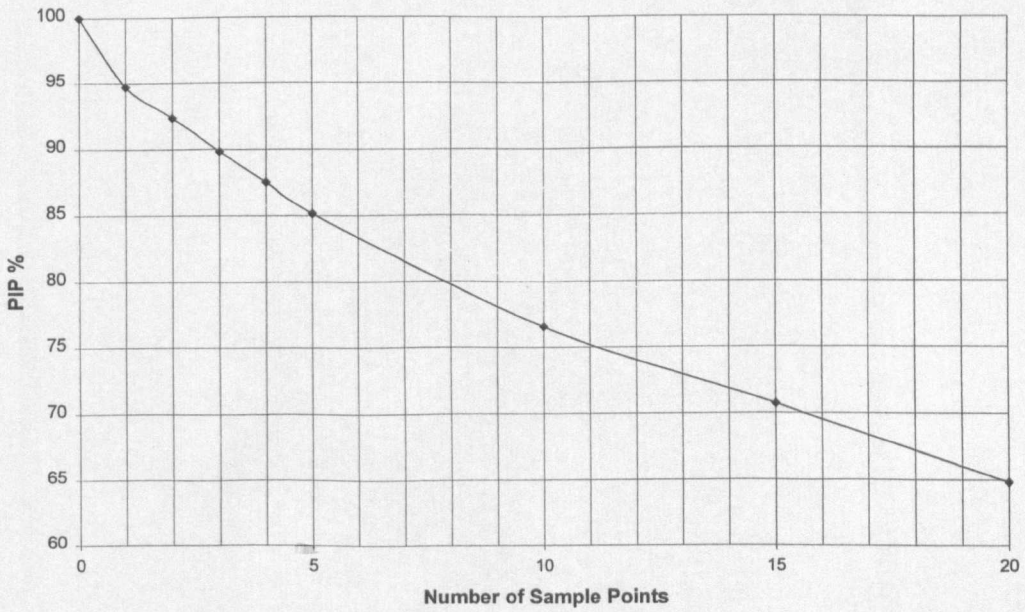

Figure 4-68: Digit Two Contour : PIP % versus Number of Samples

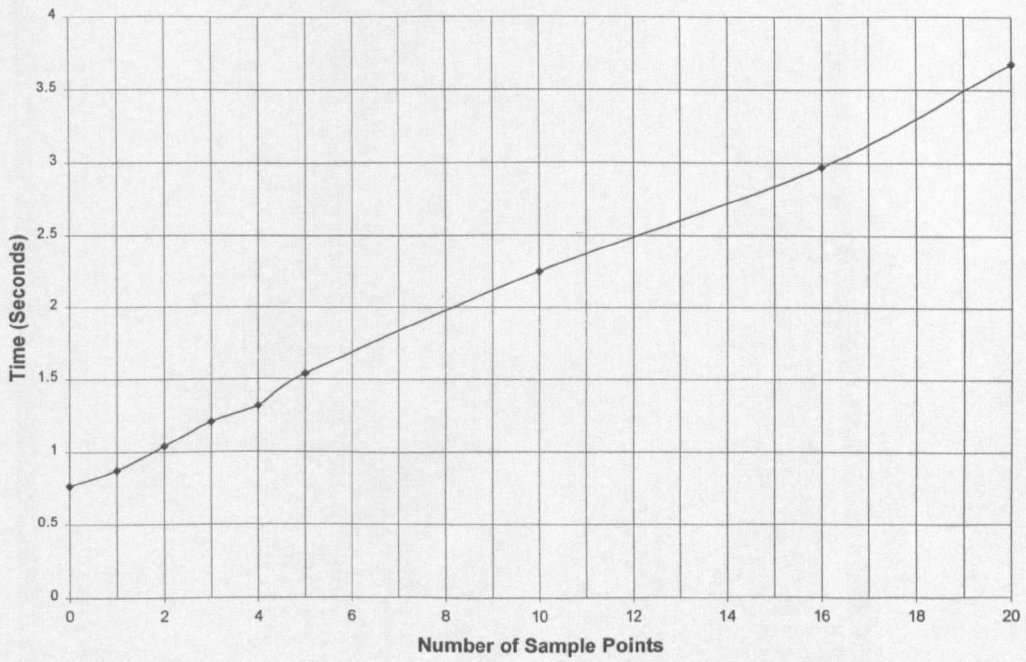

Figure 4-69: Digit Two Contour : PIP Time versus Number of Sample Points

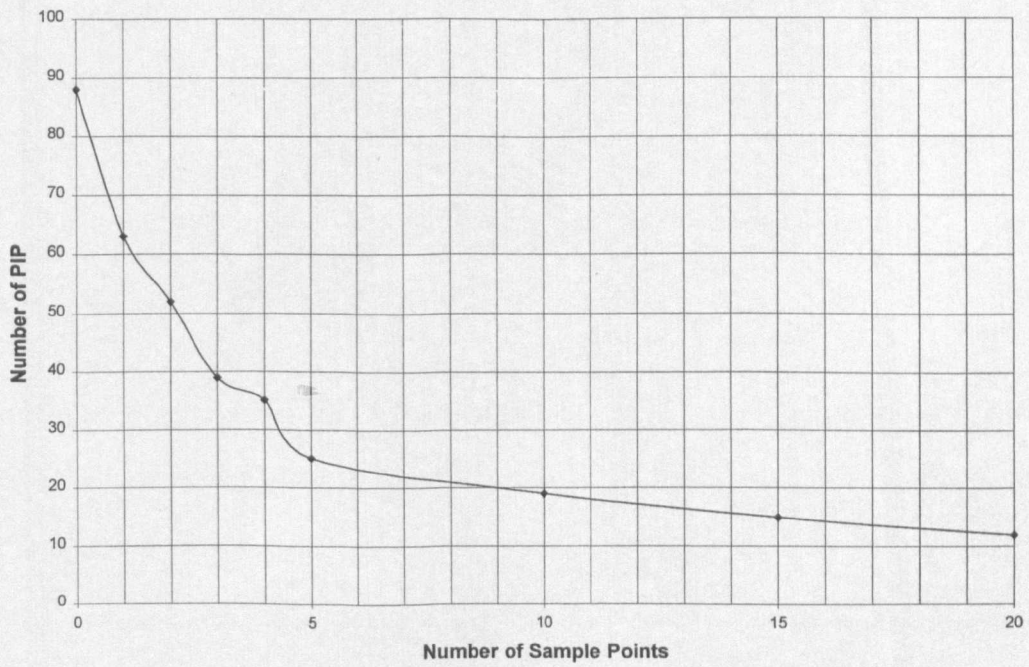

Figure 4-70: Digit Two Contour : Number of PIP versus Number of Sample Points

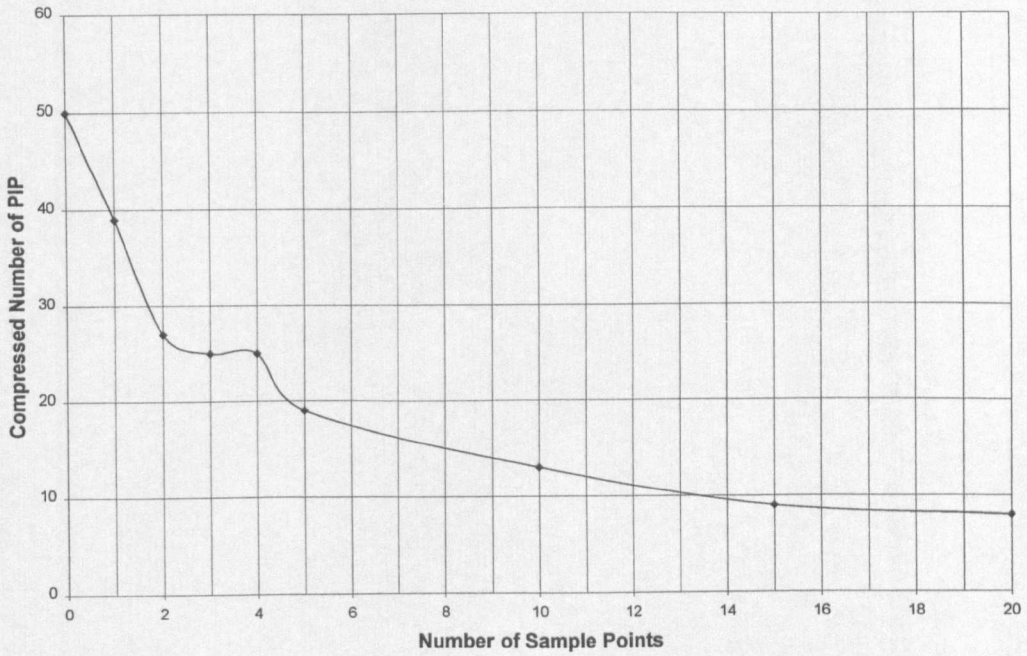

Figure 4-71: Digit Two Contour : Number of PIP (Compressed) versus Number of Sample Points

This research has shown that, for the two characteristics of the PIP identification, the new method is insensitive both to the range of spatial frequencies used to reconstruct the normalised contour, and to the range of the length of sampling points used to calculate the principal sub-histogram values. Rule 1 improves the positioning of the PIPs, especially on an elliptically shaped reconstructed contour (first spatial frequency).

These figures and graphs have been presented to show that the new technique for identifying the PIPs along a normalised reconstructed contour provides the PIP positions with an appropriate amount of 'visual' accuracy to a human observer. The 'accuracy' obtained is measured by the PIP % and the number of PIPs being approximately constant over a range of reconstruction spatial frequencies and number of sample points. This range is found not to be critical but is in the region of < 50 cycles/contour and $\leq \pm 5$ sample points along the contour. Note that the value of 25 cycles/contour and ± 5 sampling contour points are used for the contour information used by the genetic algorithm, as described in Chapter 5.

4.7 Genetic Algorithm Input

The basic PIP information (Figure 4-72) is not at this stage in the processing, in a form that is suitable for input to a genetic algorithm. This PIP information is initially in the form of normalised PIP x, y co-ordinates relative to the centroid of the contour and has to be transformed into PIP line-segment vector information as shown in Figure 4-73 for a triangular contour. Figure 4-79 shows the PIPs for a triangular contour, using rule 1, with the PIP line-segment vector positions and directions displayed on the right of the figure. The maximum spatial frequency in the reconstructed contour is 25 cycles/contour and the number of sample points along the contour is±5 contour points. The number of PIPs on the contour is 20, the PIP % equals 97.85%, and the PIP aspect ratio equals 1.14. Note, also, that the corners of the triangle have been identified as PIPs.

PIP Identification	PIP Normalised X Co-ordinate	PIP Normalised Y Co-ordinate
1	80	18
2	86	30
3	89	36
4	98	55
5	10	55
6	-2	54
7	-74	54
8	-82	54
9	-94	54
10	-98	54
11	-98	53
12	-6	-106
13	-2	-114
14	1	-116
15	2	-116
16	4	-113
17	14	-96
18	18	-90
19	25	-76
20	28	-72

Figure 4-72: PIP Co-ordinates for a Triangular Contour

PIP Identification	Start Quadrant	Finish Quadrant	Length	Direction
1	2	2	6	3
2	2	2	3	3
3	2	2	9	3
4	2	2	38	6
5	2	3	5	6
6	3	3	31	6
7	3	3	3	6
8	3	3	5	6
9	3	3	2	6
10	3	3	0	0
11	3	0	80	1
12	0	0	4	1
13	0	1	2	1
14	1	1	0	2
15	1	1	2	3
16	1	1	9	3
17	1	1	3	3
18	1	1	7	3
19	1	1	2	3
20	1	2	45	3

Figure 4-73: PIP Vector Information for a Triangular Contour

These PIP line-segment vectors provide a spatial description of the individual line-segments between PIPs along a contour. Observing these PIP line-segment vectors as groups around the contour will make it possible to identify various sections of the contour, possibly even when parts of the contour have been occluded. If this PIP line-segment vector information can be input into a genetic algorithm so that the genetic algorithm can evolve a number of ways to observe these groups of PIPs around the contour, then the 'best' ways to observe the contour can be identified.

These PIP co-ordinates are recorded and are in a normalised form, so that the position of the PIP is in a similar location for various examples of a particular contour. The PIP aspect ratio is calculated from the maximum and minimum x and y PIP co-ordinates. The value of this aspect ratio can be used as a contour feature in a recognition process if required.

The PIP information in Figure 4-73 has to be rearranged into a suitable form for use by a genetic algorithm. The PIP co-ordinates are converted into vectors, i.e. length and direction, where the

direction is resolved into one of the 8 Freeman chain code directions (0 to 7). The bounding rectangle, calculated from the maximum and minimum values for the PIPx and y co-ordinates, is also divided into quadrants. The quadrants are centred on the centroid of the normalised contour (Figure 4-79) and the processed vector PIP information is shown in Figure 4-73. Note that the start and finish quadrant information has been added to the PIP vector length and direction, and that the vector length values have been scaled to a total PIP vector length of 255 units. The quadrants are numbered from 0 to 3 in a clockwise direction starting from the top left-hand quadrant.

The PIP information, in Figure 4-73, can be further processed (compressed) to combine the individual vectors that have the same direction and are sequential in the PIP list (Figure 4-74). This compressed list of PIP vectors is treated as a cyclic list. In this way the PIP information is independent of the starting point in the list, i.e. sequential line-segment vectors will be joined over the end of the list with the vectors at the beginning of the list with the same directions. A PIP vector length of zero indicates a rounding down resolution in the integer arithmetic.

PIP Identification	Start Quadrant	Finish Quadrant	Length	Direction
1	2	3	84	6
2	3	3	0	0
3	3	1	86	1
4	1	1	0	2
5	1	2	86	3

Figure 4-74: Compressed PIP Vector Information for a Triangular Contour

PIP Identification	PIP Normalised X Co-ordinate	PIP Normalised Y Co-ordinate
1	94	0
2	96	31
3	90	46
4	79	48
5	48	47
6	24	48
7	-4	47
8	-30	48
9	-56	47
10	-80	48
11	-92	44
12	-96	27
13	-94	-1
14	-96	-31
15	-90	-46
16	-79	-48
17	-48	-47
18	-24	-48
19	4	-47
20	30	-48
21	56	-47
22	80	-48
23	92	-44
24	96	-27

Figure 4-75: PIP Co-ordinates for Rectangular Contour

Figure 4-75 to Figure 4-77 show similar information for a rectangular contour, with the PIP line-segment vector information shown in Figure 4-80.

PIP Identification	Start Quadrant	Finish Quadrant	Length	Direction
1	1	2	14	4
2	2	2	7	4
3	2	2	5	6
4	2	2	14	6
5	2	2	11	6
6	2	3	13	6
7	3	3	12	6
8	3	3	12	6
9	3	3	11	6
10	3	3	6	6
11	3	3	8	0
12	3	0	13	0
13	0	0	14	0
14	0	0	7	0
15	0	0	5	2
16	0	0	14	2
17	0	0	11	2
18	0	1	13	2
19	1	1	12	2
20	1	1	12	2
21	1	1	11	2
22	1	1	6	2
23	1	1	8	4
24	1	1	13	4

Figure 4-76: PIP Vector Information for Rectangular Contour

PIP Identification	Start Quadrant	Finish Quadrant	Length	Direction
1	2	3	84	6
2	3	0	42	0
3	0	1	84	2
4	1	2	42	4

Figure 4-77: Compressed PIP Vector Information for a Rectangular Contour

The compressed PIP vector information for a circular contour is shown in Figure 4-78 and displayed in Figure 4-81.

PIP Identification	Start Quadrant	Finish Quadrant	Length	Direction
1	1	2	15	4
2	2	2	29	5
3	2	2	16	6
4	2	2	2	5
5	2	2	2	6
6	2	3	1	5
7	3	3	21	6
8	3	3	7	7
9	3	3	3	6
10	3	3	21	7
11	3	0	30	0
12	0	0	24	1
13	0	0	1	2
14	0	0	4	1
15	0	1	29	2
16	1	1	1	3
17	1	1	2	2
18	1	1	36	3
19	1	1	16	4
20	1	1	0	6

Figure 4-78: Compressed PIP Vector Information for a Circular Contour

Figure 4-79 to Figure 4-81 show that the new PIP measurement method designed by the research in this chapter calculated the PIPs for regular geometric shapes correctly. Other techniques reported in the literature often have difficulty obtaining the PIPs for these types of regular geometric shapes.

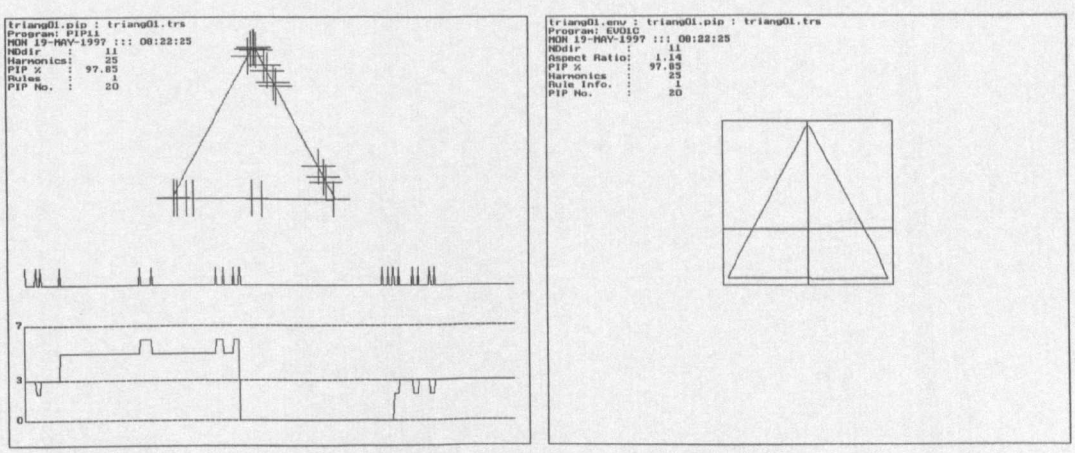

Figure 4-79: Triangular Contour PIP - PIP Positions and Directions

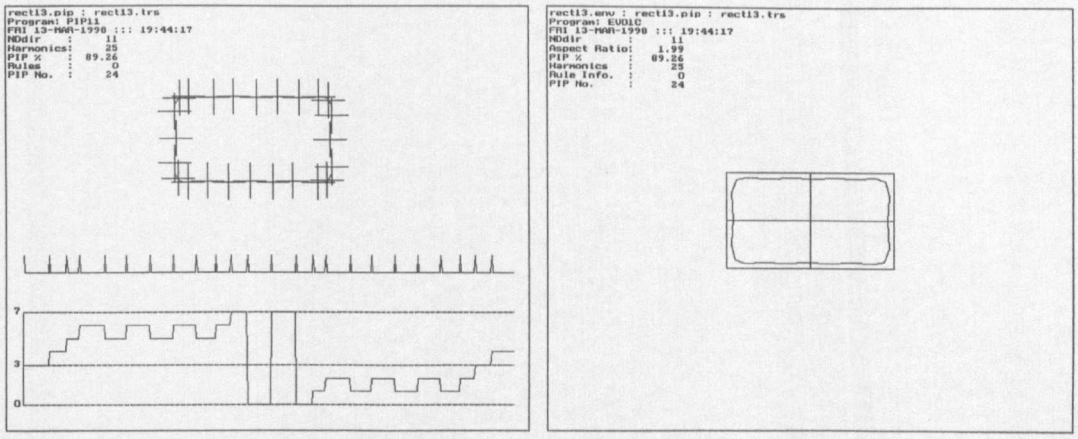

Figure 4-80: Rectangular Contour PIP - PIP Positions and Directions

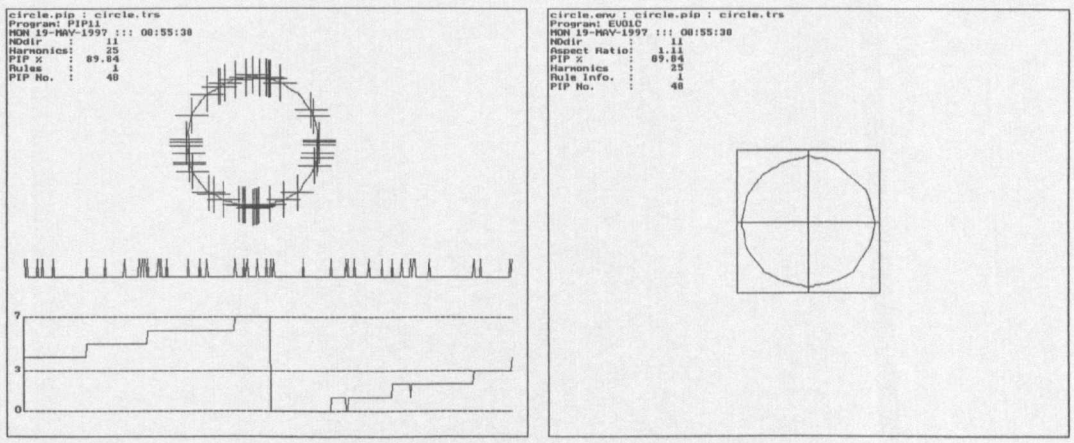

Figure 4-81: Circular Contour PIP - PIP Positions and Directions

4.8 Summary and Conclusions

The research work reported in this chapter has investigated and developed a new technique for obtaining information about a contour that is in a form suitable for use by a genetic algorithm. In order to compare contours from the same shaped object, it is essential to normalise the co-ordinates of the contour so that the comparison will be independent of translation, rotation, scale and contour starting point. The application of an affine transformation (Lu, 1997 and Ip and Shen, 1998) may be appropriate to overcome any skew present in a sequence of images. Normalisation for skew has not been considered in this research but is recommended that it should be included in any future research work.

The correction for contour start point is rarely mentioned in the literature. The relevant references either assume that this correction is applied or do not perform it at all. It is hard to understand how correct contour features can be obtained without performing the contour start point correction. Care must also be taken when normalising a contour to make sure that the normalising method correctly transfers the start point to the correct end of the first spatial frequency ellipse.

In order to improve the recognition capabilities of Fourier and moment descriptors, this research has shown that it is necessary to obtain some information about the PIPs or dominant points along a contour. The position of these PIPs, for correctly normalised contours, will most likely provide the extra information required to arbitrate in the cases where the Fourier descriptors and moments cannot provide the necessary spatial resolution to distinguish shapes which are similar but of different objects. The references studied on the topic, of identifying the positions of high curvature on a contour, all seem to concentrate on the accuracy of the methods. Some references looked at the extra information contained in a multi-scale analysis of a contour. Most methods reported in the literature require a set of parameters that usually have fairly critical values. Very few references investigated the more geometric shapes, such as squares, rectangles, triangles and general polygons.

A new method has been developed in this chapter that finds the predominant curvature changes on a contour. A multi-scale analysis of the contour is also possible with this new method, by varying the spatial bandwidth of the reconstructed contour, using a Gaussian shaped low-pass filter in the spatial frequency domain, and by varying the length of the contour sample points, which are involved in the curvature calculations. The range of values for these parameters is flexible and the PIP positions do not vary significantly within the experimental ranges used. Multi-scale analysis reported in the literature usually uses a Gaussian convolution process in the spatial domain. This new method can perform the same filtering operation in the spatial frequency domain as the contour is being normalised and hence is more efficient.

The design of a format for the contour information, which is in a form suitable for input to a recognition process or genetic algorithm, was a difficult task. Finalising the design of the contour information effectively fixed the structure of the chromosomes in the genetic algorithm and also influenced how the fitness of a chromosome is calculated. The process of matching the environmental information about a contour to the chromosome in the genetic algorithm was not trivial and, in practice, the research effort took a considerable length of time to finalise a suitable method for transferring contour PIP information into the genetic algorithm. The chromosome structure for the genetic algorithm that matches the PIP information along a contour is discussed in Chapter 5.

The research work reported in this chapter has confirmed that one of the main difficulties with all the methods used to analyse the properties of a contour is in the collection of a set of numerical data that is suitable for input into a genetic algorithm and/or recognition process. After applying the new technique for PIP identification, as described above, on a variety of contours, the results obtained have indicated that this new technique was suitable for the production of normalised PIP line-segment vectors. With the inclusion of the start and finish quadrant information these PIP line-segment vectors can provide the extra information which will enhance the recognition capabilities of the` Fourier descriptors and moment descriptors.

Multi-scale analyses reported in the literature usually use a Gaussian convolution operation in the spatial domain. This new technique allows for an efficient use of the spatial frequency normalisation process, because the Gaussian filtering can be performed in the spatial frequency domain while the contour is being normalised. The multi-scale capability of the new method during the collection of contour PIP information is suggested as a suitable topic for further research (see Chapter 9).

The research work of Chapter 5 discusses the use of these normalised PIP line-segment vectors as an input to a genetic algorithm. The chromosome in this genetic algorithm is structured to make use

of this type of input. The genetic algorithm investigated in Chapter 5 is required not just to optimise various parameters but to develop a way to examine or observe the contour PIP line-segment vectors in a variety of ways. The various combinations of these vectors can then be used to identify parts of a contour that have some distinguishing high curvature features. Different shapes must have distinguishing features along the shape boundary contour otherwise they cannot be recognised. The genetic algorithm must, therefore, develop a number of solutions to the recognition problem rather that just the 'best' solution as is usual with genetic algorithms. A number of solutions will make it possible to examine various parts of a variety of example contours so that the 'a number of best ways' to look at a contour can be evolved hence the accuracy of the recognition process will be improved.

5. Contour Shape Observation Using Chromosome Encoding

5.1 Introduction

The result of the research work described in Chapter 4 achieved a coding of a contour in the form of PIP line-segment vectors. The features of a line-segment vector are length, direction and the quadrants in which the line-segment starts and finishes. This specification for each line-segment forms an environment where each line-segment vector is defined by these features, see Figure 5-1 for a triangular contour (see also Chapter 4). The quadrants are centred on the centroid of the contour and are bounded by the width and height of the contour line-segments. The difficult problem of constructing a chromosome that can make use of this contour data is the subject of the first part of the research work discussed in this chapter. The second part of this chapter discusses how to set the crossover and mutation probabilities so that a number of solutions are evolved by the genetic algorithm. A single 'best' solution does not, in general, give sufficient information about a contour for good recognition to occur.

PIP Identification	Start Quadrant	Finish Quadrant	Length	Direction
1	2	3	84	6
2	3	3	0	0
3	3	1	86	1
4	1	1	0	2
5	1	2	86	3

Figure 5-1: Line-Segment Encoding for a Triangular Contour

A standard or canonical genetic algorithm is investigated that can use this type of chromosome to evolve a 'way of looking at or observing' part of a contour. The parameters of this genetic algorithm are not adjusted to obtain just the 'best' solution, but are varied to investigate suitable values that evolve a number of good, but less fit solutions. A fitness dispersion measure, from Lis (1995), was studied, which was used to identify the diversity in the population. The research described in this chapter presents a new diversity measure that examines the bit-patterns in the chromosome to indicate the extent of the population diversity. The bit-patterns chosen are those that control the way in which the chromosome observes the contour (see observation genes, Figure 5-2). Both of these

measures are investigated in the research to find out if they are suitable to identify the best values for the crossover and mutation probabilities that would evolve a number of reasonably fit solutions.

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
1	Start Quadrant	Search Direction (Clockwise or Anticlockwise)	Number of line-segments	Start search position	Line-segment length threshold
2	2 Bits	1 bit	3 bits	5 bits	5 bits
3	00[0]	1[1]	100[4]	00111[7]	00011[3]
	Observation Genes			**Parameter Genes**	

Figure 5-2: Chromosome Structure

The research investigated the form of the fitness function and typical results for a variety of experimental tests (Figure 5-30) are presented as evidence of the behaviour of the standard genetic algorithm as the crossover and mutation probabilities are varied. The 'Figure Ident.' in the second column is a shortened version of the date on which the data was collected. The concept of unique (individual) solutions, i.e. chromosomes with a unique bit-pattern, and different solutions, i.e. chromosomes with observation genes that define different ways to observe a contour, are discussed in relation to the various values of the crossover and mutation probabilities.

5.2 Review of Related Work

The genetic algorithm has been applied to image processing mainly in the following areas: Model Matching, Template Matching, Morphological Operators, Image Segmentation, Edge and Region Detection, Image Restoration and Enhancement, and Parallel Processing. Figure 5-3 shows the range of Image Analysis topics using genetic algorithms to perform the analysis. In general, these papers describe various ways to set up the parameters of a genetic algorithm to find the 'best' solution for the analysis. The chromosome used in these genetic algorithms usually encodes parameters of such processes as filters, shape models and morphological operators.

Related Work	References
General Optimisation Problems	Amaral, 1995; Bandyopadhyay and Pal, 1997.
Model Matching	Brunnstrom and Stoddart, 1996; Das, 1996; Hata, 1996; Hill and Taylor, 1991; Hsieh, 1996; Hung and Adeli, 1994, Ozcan and Mohan, 1997; Saito and Mori, 1996; Toet and Hajema, 1995; Tsang, 1997; Wang and Funakubo, 1996.
Template Matching	Bhattacharjya and Roysam, 1994; Roth and Levine, 1994; Sakano, 1996; Wilson, 1985.
Morphological Operators	Bala and Wechler, 1993 and 1996; Brockett and Maragos, 1994; Loncaric and Dhawan, 1995.
Image Segmentation	Andry and Tarroux, 1994; Bhanu, 1995; Jacquelin, 1997; Scheunders, 1996.
Edge and Region Detection	Bhandarkar, 1994, Chun and Yang, 1996.
Image Restoration and Enhancement	Chen, 1996; Grefenstette, 1986; Pal, 1994; Tian and Tsui, 1996.
Parallel Processing	Bhattacharjya and Roysam, 1994; Chen, 1996; Fogarty and Huang, 1990; Hung and Adeli, 1994.

Figure 5-3: Image Analysis Using the Genetic Algorithm References

Hill and Taylor (1991) described the application of a genetic algorithm to model-based image interpretation. The space of possible model parameter values was searched and model instances were projected back into the image until one was found which was consistent with the observed image. The model was instantiated by choosing values for a set of 6 shape parameters and 4 transformation parameters. These parameters of the model were encoded in the chromosome as unsigned gray-code binary integers. This paper pointed out that "if the genetic algorithm could be employed to extract multiple optima, the functionality of the method would be greatly enhanced. The purpose of the search would no longer be to locate the single model instantiation most likely to be the best match to the object (a left ventricle in a heart), but would be to extract a handful of strong, ranked candidates for the object. The approach adopted was to reduce the number of individuals in over-crowded areas of the search space by modifying their function values using a 'crowding factor' (see also 'sharing' in Chapter 2).

Liu and Huang (1998) proposed a hybrid pattern recognition system based on evolutionary algorithms and fast simulated annealing. The proposed method relied on a polynomial approximation technique that extracted approximate feature vectors characteristic of a given shape. Shape recognition was implemented as a matrix based classifier. The matrix was defined as the matching states between a model graph and a scene graph, where the elements of the matrix were the possible matches. The Boltzmann distribution in the simulated annealing algorithm was replaced by the Cauchy distribution in order to reduce the probability of incorrect optimisation during the local search process. Experimental results, which included overlapping objects, were presented to show the effectiveness of the proposed algorithm.

Saito and Mori (1996) proposed a method that evaluated the similarity between a model and an image at each view angle. A genetic algorithm found the model that had the maximum evaluation. In this proposed method, a *sharing* scheme was used for finding multiple solutions efficiently. "By *sharing*, evaluation of the string is decreased if the string is similar to other strings in the population. This prevents the one point convergence of the solutions and then the multiple solutions can be obtained."

Kawaguchi and Nag (1998) presented a new genetic algorithm to extract and locate partially visible objects from a 2D image. Prominent fragments from the boundary of a model object were matched to those of an image object by the genetic algorithm. A double maximisation problem was solved in which an inner process found the best vertex correspondence between image and model fragments and an outer process identified image and model fragments that had the best match. This new genetic algorithm also found the location of objects with distorted boundaries.

Toyama (1998) proposed a method of model-based pose estimation using a genetic algorithm to optimise the model parameters. A new concept of fitness was described that takes edge-directions in the input image into consideration. Four edge-direction images were detected from the input image

and the fitness was calculated using matching information from all four edge-direction images. Elitism was used during the genetic algorithm selection process and a steepest descent method was used as a local search. Results were presented that showed the estimated pose produced by the genetic algorithm was accurate.

Tsang (1997) presented a technique for affine invariant recognition of single near planar object shapes from broken boundaries. Matching scores between pairs of object contours were calculated using a combination of a genetic algorithm and a distance transform. A dominant point technique was used to calculate a global and a local similarity between two object shapes. "The problems and limitations in dominant point techniques can be attributed to the fact that they are largely based on local boundary information that can be easily destroyed by distortion and noise contamination in practice." This paper reported on the use of a genetic algorithm to determine the affine transform that results in the best alignment between the reference and the test shapes.

Roth and Levine (1994) discussed the extraction of geometric primitives from geometric sensor data. A minimal subset is the smallest number of points necessary to define a unique instance of a geometric primitive. A genetic algorithm based on a minimal subset representation was used to perform primitive extraction as an optimisation problem. The chromosome representation used by the authors was not the usual bit string representation, but a representation of the geometric primitive by the minimal subset of defining points. It was shown that this genetic approach is capable of extracting more complex primitives than the Hough Transform (see Leavers, 1992). "The advantages of the genetic algorithm over the Hough transform are that it can be applied to a much wider variety of primitives and that it can be easily parallelised. One simple approach to parallelisation is to concentrate on cost function evaluation, which can be implemented on a wide variety of parallel architectures. The Hough transform, by contrast, is much more difficult to parallelise."

Wilson (1985) showed that a certain model of the primate retino-cortical mapping 'sees' all centred objects with the same 'object-resolution', independent of apparent size. A system was described which permits recognition of patterns using templates in a cortex-like space. It was suggested that with an adaptive production system, such as the classifier system described in Holland (1987), the recognition process could be made self-organising.

Mirmehdi (1997) investigated the optimisation of the image low-level feature extraction chain by using a genetic algorithm. "The transformation of signals into symbols is critically important if higher levels of image recognition are to use them as building blocks for scene interpretation." The fitness function was a performance measure that reflects the quality of an extracted set of features and consisted of three major steps: generate the edge map, detect line-segments and measure the quality of the lines. "The quality measure acted on the hypothesised boundaries between regions and gives quality values that reflected the statistical distribution of pixels in the area local to the hypothesised line." The method described in this paper determined an optimum parameter set rather than selecting individual parameters for a single processing step. Emphasis was placed on "the importance of having a dependable set of features in the early stages of vision that will serve as a solid building block for the higher level stages."

Srinivas and Patnaik (1994) discussed adaptive crossover and mutation probabilities and recommended the use of adaptive probabilities to realise the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the genetic algorithm. The crossover and mutation probabilities were varied depending on the fitness values of the solutions. High fitness solutions were 'protected' while solutions with sub-average fitness are totally disrupted. It was suggested that, by using adaptively varying probabilities for crossover and mutation, a solution to the problem of deciding the optimal values for these probabilities is provided. An approach by Whitley and Starkweather (1990) was referenced in this paper which dynamically varied the mutation probability according to the Hamming distance between the parent chromosomes. The

diversity of the population was sustained by subjecting similar solutions to increased levels of mutation. Extensive experiments have been conducted, by the authors, on a wide range of problems including TSP, neural network weight-optimisation, and generation of test vectors for Very Large Scale Integrated (VLSI) circuits. In most cases, the adaptation of the crossover and mutation probabilities enabled their adaptive genetic algorithm to outperform the standard genetic algorithm. Specifically, the authors have observed that the adaptive genetic algorithm performed very well for problems that are highly epistatic and multi-modal.

Katz and Thrift (1994) directly applied a set of linear operators (filters) to the image pixel intensity data. A genetic algorithm was applied to a population of filters (with real coefficients) and these filters were then used to distinguish targets from background clutter. This paper noted that "although chromosomes represented by strings of 0's and 1's (or other low cardinal alphabets) are considered 'standard', many authors, e.g. Davis (1991), used strings of real numbers (or other high cardinal alphabets) as the chromosome representation."

Davis (1991) reported empirical success with high cardinal alphabets, even though schema theory, which was developed for binary alphabets (Holland, 1992), pointed to advantages for low-cardinality chromosome coding. This paper referred to Goldberg (1990) where a theory based on virtual alphabets is presented to explain the success of high-cardinality chromosome coding and notes, "Selection typically dominates early genetic algorithm performance in these cases and results in the discovery of a virtual low-cardinality alphabet, which then dominates genetic evolution."

Chen (1996) presented a parallel genetic algorithm based on the island model for image restoration. The algorithm divided a large population into smaller sub-populations and executed the main loop of a traditional genetic algorithm on each processor with its own sub-population in parallel. The simulation results showed that the algorithm achieved a linear speed-up with the number of processors. The parallel algorithm was also shown to have a better performance on image

restoration than the traditional genetic algorithm. Performance degradation due to uniformity of a sub-population was avoided by selecting the best individual from each sub-population and moving it to another sub-population. The worst individual was replaced by the winning individual from the adjacent sub-population. This migration between islands used a *ring* topology and the migration interval is k generations.

Anelli (1998) presented a new stochastic approach to the task of the decomposition of arbitrarily shaped binary morphological structuring elements using a genetic algorithm. No constraints were imposed on the shape of the initial structuring element and no assumptions were made on the elementary factors that were selected within a given set. The main purpose of the work presented in this paper was to develop a method that would be able to give a preliminary solution to the problem of '*optimal* decomposition of non-convex structural elements into concatenations of generic elementary operations'. Due to the extremely high computational load and large memory requirements required by this approach, further research was suggested with a parallel processing environment using the Message Passing Interface (MPI) protocol.

Chai and Ma (1998) formulated epipolar geometry estimation as a global optimisation problem and presented a genetic algorithm that searched for the optimum parameters. This paper noted that many computer vision algorithms assumed that the least squares method was sufficient to deal with data corrupted by noise. In many applications the data was not only noisy but contained outliers, i.e. data that was in gross disagreement with a postulated noise model. Results were presented to show that the proposed genetic algorithm reduced the effect of these outliers for this application.

Most of the genetic algorithm studies reported in the above referenced literature used the genetic algorithm to adjust a set of parameters that are optimised to find a 'best' solution to a particular problem. Binary and real (floating point) valued chromosomes are used and the genetic algorithm parameters, such as crossover and mutation probabilities, are chosen to find a single best solution.

For example, a template may be matched to the contour of an object with a minimum Euclidean *rms* error between the template and the contour. This *rms* error would be used as a fitness measure. This type of fitness function is often calculated as an *rms* error between the desired result and the result produced by the genetic algorithm. Techniques for termination of the genetic algorithm are also discussed in the literature and usually consist of a termination criterion dependent on the change in the fitness reducing below a predefined threshold.

Spears (1997) considered novel methods for evolving species in a standard generational evolutionary algorithm. Unlike other methods, the proposed techniques replaced the concept of distance between individuals (see Goldberg and Richardson, 1987) with *tag* bits that identify the species to which an individual belongs. Similarity would then become simply a matter of checking if two individuals had the same label (tag). Results, so far, suggested that the added precision of the distance metric was often not required and that restricted mating and sharing with labels was an efficient technique. It was noted that this work had similarities to the evolutionary algorithm research performed on parallel architectures.

The standard or canonical genetic algorithm has been chosen for the research work in this chapter because it can be analysed by the schema theory (see Chapter 8) and therefore can be shown to be behaving as expected. The schema chosen for analysis are the genes themselves. The chromosome designed in this chapter encodes the contour shape features, i.e. how to observe the contour to produce a high fitness value rather than any parameters for contour models and templates. The genetic algorithm designed in this chapter has to evolve a number of less fit solutions so that various parts of a contour can be observed using a variety of 'good' chromosomes. The genetic algorithms in the literature may be satisfied with less fit solutions if the 'best' solution takes too long or is too difficult to evolve but the genetic algorithm developed in this chapter must have a number of solutions to enable good recognition to be achieved.

5.3 Chromosome Encoding

The design of the chromosome encoding used in the research described in this chapter takes a different approach to the above referenced literature. The encoding has been chosen such that the chromosome defines the way in which any particular part of the contour is to be examined or observed, rather than encoding parameters of the contour or contour model directly. The genes in the chromosome identify where to start looking at the contour (genes 1 and 4), whether to search clockwise or anti-clockwise (gene 2) and how many line-segments to include in the analysis (gene 3). The chromosome also specifies which line-segments are to be ignored by using a minimum length threshold (gene 5) (Figure 5-2, where row 3 shows example values).

The chromosome is thus encoded in the following way:

1. A gene to identify the start quadrant for the sequence of line-segments (gene 1).
2. A gene to specify whether the line-segments will be grouped in a clockwise or anti-clockwise direction (gene 2).
3. A gene to define the number of line-segments to use (gene 3).
4. A gene to indicate where to start looking in the list of line-segments for a contour (gene 4).
5. A gene to provide a threshold on the minimum length of a line-segment to ignore in the fitness calculation (gene 5).

Genes 1, 2, and 3 are the observation genes, i.e. they specify how to observe the contour. Genes 4 and 5 are parameter genes, i.e. the value affecting the operation of the observation genes.

Two types of encoding values are used:

1. A binary chromosome with sixteen bits.
2. An integer chromosome with five integers, one for each gene.

These two types of chromosome are used in order to investigate the most appropriate format for

encoding the gene in the chromosome. The canonical genetic algorithm (see Holland, 1986 and 1992) used a binary chromosome whereas Michalewicz (1992) developed the floating-point representation of a chromosome. The integer version of the chromosome was chosen in this research, to investigate if the behaviour of the genetic algorithm, in any way, follows or improves upon that of a genetic algorithm using a binary chromosome.

The total number of combinations of gene values for this sixteen-bit chromosome is, therefore, 65536. The start quadrant is encoded in two bits and the direction of search is encoded in one bit. The number of line-segments is coded in three bits, usually scaled from two to eight. The start point, for searching the list of contour line-segments, is coded in five bits and is usually scaled as a fraction of the length of the input line-segment list (see Figure 5-2). The line-segment length threshold is coded in five bits and is usually scaled from 0 to 20. The integer version of the chromosome uses a separate integer value for each of the genes described above. The choice of scaling for the genes will be discussed later in this chapter.

A particular chromosome, therefore, specifies the way in which the line-segments of the contour are to be examined or observed. The example chromosome, shown in the bottom row of Figure 5-2, encodes the following statement:

Start looking in the input contour line-segment list(see Figure 5-1) *at the point 25% (7/31) from the start of the list. Find a line-segment that starts in quadrant 0 and calculate a fitness for 4 line-segments in a clockwise direction, noting in which quadrant each of the 4 line-segments starts and finishes. Ignore a line-segment whose length is less than the threshold value of 3.*

Contours, which are slowly varying with time, may also be analysed by this type of chromosome encoding. The genetic algorithm is ideally suited for a parallel implementation and could possibly be trained on the various contours as the object moves in the image. The group of fit chromosomes

would change with time and would develop to match the contour as it changes.

5.4 Genetic Algorithm Design

The chromosome used in this research had a fixed length. The genetic programming methods investigated in Chapter 3 usually involved variable length chromosomes. Difficulties occur, in practice, when the length of the chromosome becomes too large for the particular processor being used. Artificial rules are normally introduced to truncate the length of the chromosome. This process suffers from many problems associated with the removal of genetic information, which may be required at a later stage in the evolution of a solution. A fixed length chromosome appears to be a more 'natural' way of encoding the genetic information.

The pseudo code for the genetic algorithm, used for the experimental tests described below, is shown in Figure 5-4. The genetic algorithm is of the canonical or standard form, with roulette wheel selection (i.e. probability of selection is proportional to the fitness of the chromosome) of chromosomes for the application of the standard crossover and mutation operators. One point crossover is used and mutation is applied with the specified probability to each bit in the sixteen-bit chromosome. Mutation, when the chromosome is five integers, is applied as a randomly selected positive or negative small change to the gene. Population replacement is used with random chromosome replacement for some of the test cases.

Choosing the most appropriate values for the crossover and mutation probabilities requires careful consideration. A range of solutions is required from the genetic algorithm rather than just the best solution. When trying to recognise a contour, it would seem that better recognition should be achieved when the contour is observed in many different ways. Looking at a contour in only one particular manner may not cover all variations of the contour's shape, which would occur for the many examples of a shape as presented to an algorithm for recognition.

```
Genetic Algorithm Pseudo Code

Input Training Set Line-segment Environment
Choose Binary or Integer Chromosome Type
Set chromosome fitness-threshold value
Initialise Algorithm Parameters
Initialise Random Population of Chromosomes

For-each-generation

For-whole-population
  Calculate fitness of each chromosome
End-for-whole-population

Sort chromosomes according to their fitness value

For-each-chromosome-with-fitness > fitness-threshold
  Record each training set line-segment mean and standard
  deviation value as specified in the selected chromosome genes
End-for-each-chromosome-with-fitness > fitness-threshold

For-whole-population
  Calculate-record-and-display
    Chromosome fitness histogram
    Chromosome diversity
    Unique solutions histogram
    Number of unique solutions
    Average Population fitness so far[on-line performance]
  End-calculate-record-and-display
End-for-whole-population

If  (STOP)  condition then EXIT
End if

Produce-a-new-population
  Repeat
    Either
    Select a pair of parent chromosomes by Roulette Wheel  Selection
    Create a pair of child  chromosomes by one point crossover and
     mutation
    Or
    Create a random pair of child chromosomes
    End either-or
  Store child chromosomes in new population
  Until new population is complete
End-produce-a-new-population

Transfer new population to population for next generation

End-for-each-generation

Display final summary statistics
Record unique solutions histogram
```

Figure 5-4: Genetic Algorithm Pseudo Code

The chromosomes are sorted on increasing fitness before each generation of the genetic algorithm. Some references, e.g. Chambers (1995, page 43), commented that the chromosomes should not be sorted before the genetic algorithm is applied, because the roulette wheel selection will be biased to the fitter individuals. The experimental tests performed in the research in this chapter do not show

any particular change in behaviour when the population is not sorted. Sorting the population before the application of the genetic operators has the advantage that the fitness information is displayed in a much more suitable manner for visual monitoring and analysis. Sorting the population also enables chromosome ranking to be easily applied if required, and the first and last chromosomes in the population are also easily identified when sorted as the maximum and minimum fitness values. These values are required for fitness scaling which can be used to improve the spread of fitness values more evenly throughout the population.

Most of the genetic algorithm references, reviewed above, described research into optimisation problems, where solutions other than the best may not necessarily be suitable as solutions to the problem. In the current research application the less fit, but good, solutions are required so that various ways of looking at the contour can be found which will eventually specify the best way to examine the contour as a whole. The best ways to examine the contour are those which identify the combination of line-segments that sufficiently describe the contour so that recognition of other instances of the contour can be achieved.

The range of solutions with the lowest fitness can also be used to specify how to look at the contour such that the combination of line-segment vectors does not match. This group of chromosomes identifies the various sections of the contour over which there is very little matching amongst the set of examples. The fitness of a chromosome is scaled between 0.0 and 1.0, therefore, evolving a population with (1.0 – calculated fitness) as the measure of the best chromosome will develop the various ways of looking at the contour which will show a mismatch over the training set of contours. Using these combinations of line-segments with the group of most fit chromosomes will add extra information on which to decide whether a contour can be recognised. A parallel version of the genetic algorithm could evolve chromosomes with the 'best' and 'worst' fitness concurrently and, thus, has the extra evidence for recognition available at the same time.

Observing a contour in many different ways may also have advantages should the contour be partially occluded. Partially occluded contours will have various combinations of line-segments available for analysis by the chromosomes. If a sufficient number of line-segments are available, then recognition may still be possible. In this case, recognition may be achieved by observing the line-segment information, which is not occluded, using only part of the many ways to look at a contour, which have been evolved by the genetic algorithm. The occurrence of partial and/or total occlusion may also be detectable using the various combinations of line-segments that have been developed by the genetic algorithm, i.e. detection of missing line-segment combinations may be possible.

The standard or canonical form of the genetic algorithm was chosen for analysis because the research in this chapter investigated the ability of the genetic algorithm to evolve various ways of observing a contour. The research did not study various possible modifications to the standard genetic algorithm itself to achieve some form of optimisation.

A number of references, e.g. Michalewicz (1992, page 77) and Davis (1991, page 64), have also suggested having real genes rather than the standard binary genes. Rather than investigate real genes, it was decided during the research work to include a study of the behaviour of an integer version for the chromosome in order to find out if the integer form had any advantages. Therefore all experimental tests were performed using both types of chromosome for comparison (see Section 5.8).

Many of the genetic algorithm references reviewed above presented graphical output showing how the average or maximum population fitness varies with each generation. Most of the analysis in these references was taken up with the performance of the genetic algorithm in being able to find the 'best' solution to the problem, e.g. investigating different ways to select chromosomes for reproduction. The referenced genetic algorithms also discuss various modifications to the genetic

algorithm that are considered to improve its performance. The research work presented in this chapter concentrated on the choice of crossover and mutation probability values that will enable the genetic algorithm to evolve a number of solutions rather than just evolving the 'best' solution.

5.5 Chromosome Fitness Calculation

The research in Chapter 3 identified that the calculation of the fitness of a chromosome, which is operating in a two-dimensional spatial environment, presents certain difficulties. The calculation may be such that the fittest chromosomes do not describe the environment correctly, i.e. are non-linear and/or asymmetric about the fittest value. The chromosome encoding described above must be protected from this problem when examining the contours from two shapes. When parts of the contour are similar, the fitness calculation should give a high value and when their shape is dissimilar, the calculation should provide a low fitness. These problems are discussed further in Chapter 7.

The objective function for the fitness, which is used for the experiments described in this chapter, is of a triangular form (see Figure 5-5). The slope of the triangular fitness function is defined by the number (n) of standard deviations for the appropriate line-segment feature, see CD in Figure 5-5. The value of n for these experiments was set to 1.0. A fitness value is calculated for each individual feature (e.g. direction of the line-segment, start quadrant) in the sequence of line-segments. The distance of a measured-value (V) from the mean (μ) for that feature modulated by the triangular objective function is output as the fitness for that feature. The distance CE, in Figure 5-5, measures the difference between the measured-value and the mean-value of a line-segment feature. The fitness value F is calculated as described by the equations in Figure 5-6. The maximum fitness, 1.0, is measured by the distance AC in Figure 5-5. The value for the number of standard deviations used by the objective function is input as a parameter to the genetic algorithm.

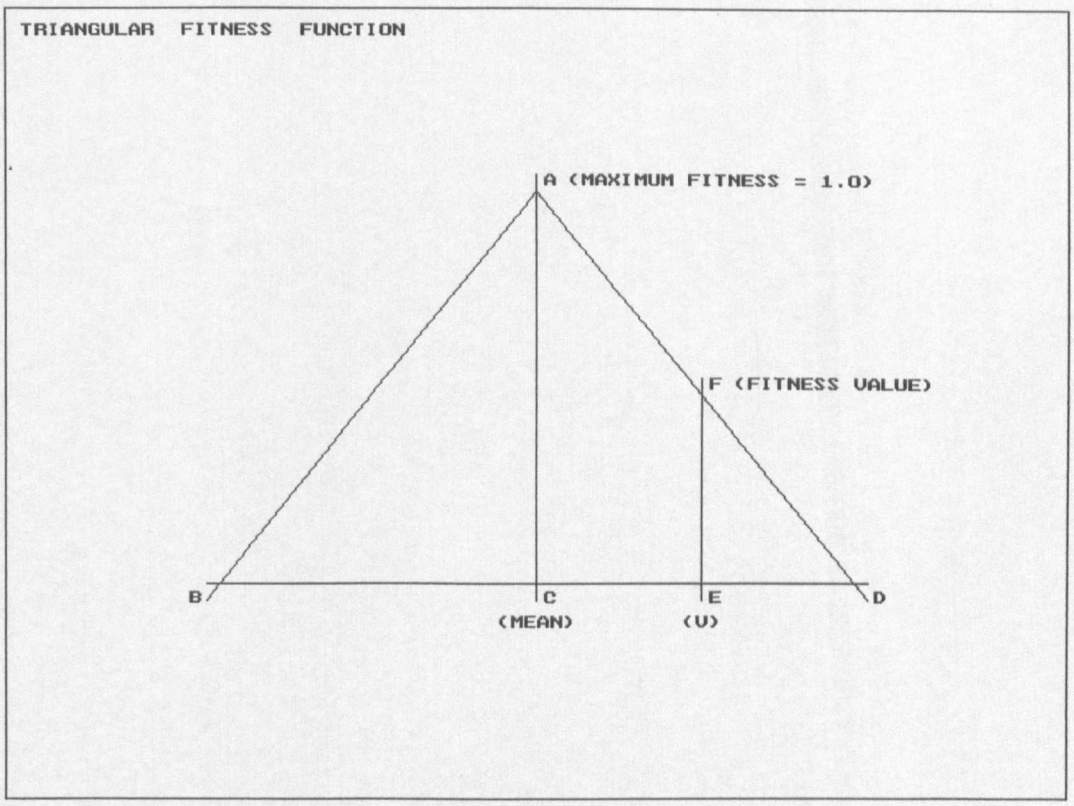

Figure 5-5: Triangular Fitness Function Definition

This particular fitness function was chosen for the research work so that the mean and standard deviation of the exemplar line-segment vectors could be taken into account. Line-segment vectors with component values contributing to a large standard deviation would, therefore, be allocated a relatively low fitness value. Line-segment vector values near to the mean would have correspondingly larger values.

$$AC = 1.0$$

$C = \mu$, where μ = mean of line-segment feature

$CD = n\sigma$, where σ = standard deviation of line-segment feature
and n = number of standard deviations to use

$CE = abs(\mathrm{V} - \mu)$, where V = value of line-segment feature

$$\frac{FE}{ED} = \frac{AC}{CD}$$

$$fitness = FE = \frac{AC.ED}{CD} = \frac{1.0(CD - CE)}{CD}$$

$$fitness = 1.0 - \left(\frac{CE}{CD}\right)$$

$$fitness = 1.0 - \left(\frac{abs(V - \mu)}{n\sigma}\right)$$

If $(\sigma < 0.01)$ then $\sigma = 0.01$, which will avoid a divide by zero.

Figure 5-6: Triangular Fitness Function Equations

An example calculation, for one line-segment length feature is given below. The equations used to calculate the fitness value are, as mentioned above, shown in Figure 5-6 and the example values are specified in Figure 5-7:

Line-Segment Feature	Value
Standard Deviation[σ]	3.0
n	1.0
Mean[μ]	6.0
Input Value[V]	8.0

Figure 5-7: Example Values for a Single Line-Segment Feature

$$n = 1.0 \tag{5.1}$$

$$\sigma = 3.0 \tag{5.2}$$

$$CD = n\sigma = 3.0 \tag{5.3}$$

$$CE = abs(V - \mu) = abs(8.0 - 6.0) = 2.0 \tag{5.4}$$

$$\begin{aligned} fitness &= 1.0 - \left(\frac{abs(V - \mu)}{n\sigma} \right) \\ &= 1.0 - \left(\frac{2.0}{3.0} \right) = 0.33 \end{aligned} \tag{5.5}$$

Hence the example fitness for this line-segment length feature, defined in Figure 5-7, equals 0.333.

A training set of contour line-segments, from examples of the same shape, was used to calculate the mean and standard deviation for each line-segment feature (see Figure 5-9), for the whole of the training set, which is examined by the chromosome. Example line-segment mean and standard deviation data for two chromosomes (1 and 16) from genetic algorithm Test 1 are shown in Figure 5-10 (see also Chapter 6). This figure lists the data for two line-segments with the mean data at the top of the list and the standard deviation data shown at the bottom of the list. An example of the training set for the digit two is shown in Figure 5-8.

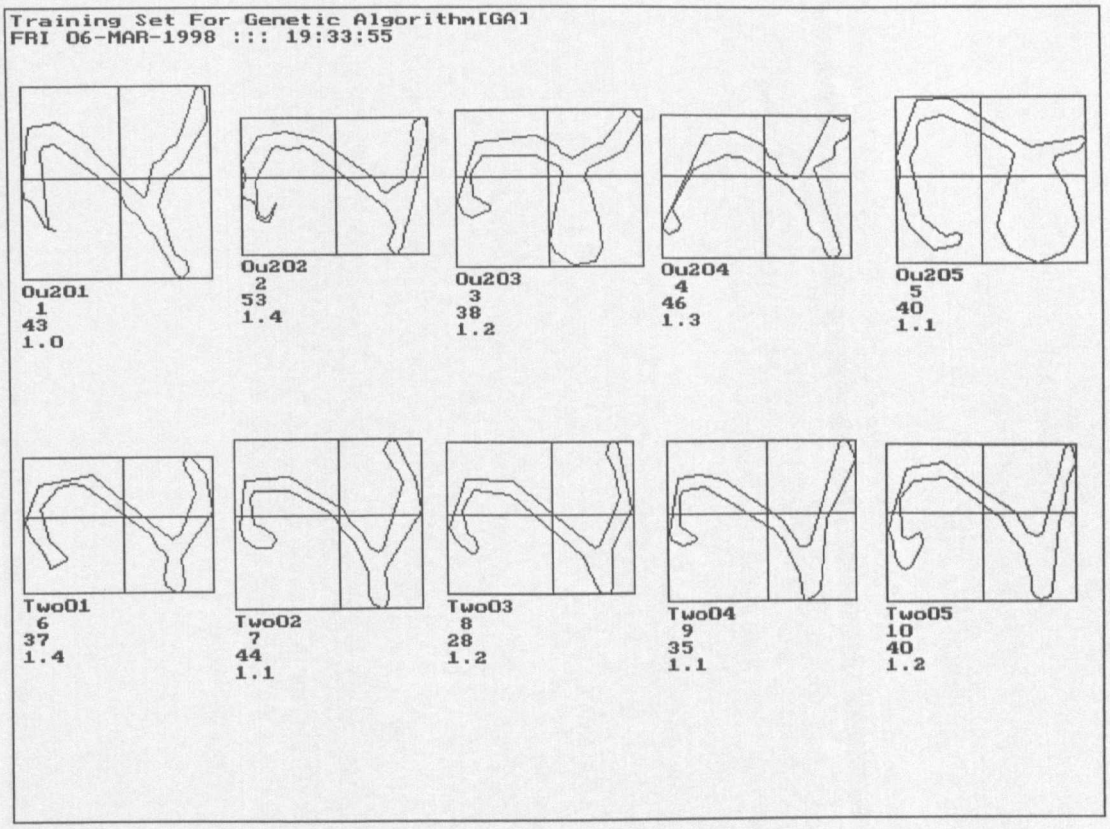

Figure 5-8: Training Set : Digit Two

The training set may contain standard or model contour shapes when required to calibrate the mean and standard deviation values for a particular shape. The model matching techniques described in the references (see Section 5.2) adjusted the parameters of the model to evolve a solution to the matching problem. The model matching described in this research evolved the best description of the model shapes contained in the training set, where this description was specified by the chromosome as the 'best' way to observe the model shapes in the training set.

Line-Segment Features
Line-Segment Length
Line-Segment Direction
Line-Segment Finish Quadrant
Line-Segment Sequence Finish Quadrant

Figure 5-9: Line-Segment Features

Generation 17

```
        Chromosome          Fitness
 1: 0   0   2   0.19   2     0.809
    Mean Data
       0.00       -- Sequence Finish Quadrant
    0.00   0.00 -- Quadrant
    9.60  32.50 -- Length
    6.00   7.00 -- Direction
Standard Deviation Data
       0.00       -- Sequence Finish Quadrant
    0.00   0.00 -- Quadrant
    4.25  11.87 -- Length
    0.00   0.00 -- Direction
```

Generation 18

```
        Chromosome          Fitness
16: 0   0   2   0.90   1     0.670
    Mean Data
       0.00       -- Sequence Finish Quadrant
    0.00   0.00 -- Quadrant
   10.00  32.90 -- Length
    5.60   6.40 -- Direction
 Standard Deviation Data
       0.00       -- Sequence Finish Quadrant
    0.00   0.00 -- Quadrant
    4.07  11.84 -- Length
    1.20   1.80 -- Direction
```

Figure 5-10: Chromosome Mean and Standard Deviation Data (GA Test 1)

The pseudo code for the chromosome fitness calculation used in the current experiments, is shown in Figure 5-11. The pseudo code for the calculation includes all the line-segment features and uses a weighting factor of unity for each fitness value in the calculation. The linearity and symmetry of this type of fitness calculation is discussed in Chapter 7.

```
Training Set Fitness Calculation Pseudo Code

For-each-chromosome

For-each-example-of-line-segment-sequence-in-training-set
  Calculate the mean and standard deviation for each
  line-segment feature
End-for-each-example-of-line-segment-sequence-in-training-set

For-each-example-of-line-segment-sequence-in-training-set
  Calculate the fitness using the triangular objective
  function for each line-segment feature
End-for-each-example-of-line-segment-sequence-in-training-set

Average the fitness values for each line-segment feature
in the whole training set

Fitness = Sum(and then average) all of the average fitness
values for each feature with unity weighting, scaled
0.0 to 1.0

If (Chromosome Looking for a Match)
  then Chromosome-fitness = Fitness
If (Chromosome Looking for a Mis-Match)
  then Chromosome-fitness = 1.0 - Fitness

End-for-each-chromosome
```

Figure 5-11: Genetic Algorithm Fitness Calculation Pseudo Code

A fitness value was, therefore, obtained for each of the line-segment features (length, direction, and finish quadrant) for every line-segment in the sequence. A fitness value was also calculated for the finish quadrant for the whole sequence of line-segments. This fitness will be relative to the mean and standard deviation for each line-segment feature, which is calculated from the whole of the training set. For the example chromosomes, Figure 5-10, 2 line-segments were observed. Hence, fitness values are calculated, using the triangular fitness function, for the 2 line-segment lengths, the 2 line-segment directions the 2 line-segment finish quadrants and 1 line-segment sequence finish quadrant. The average fitness for each of these line-segment features was then calculated, i.e. an average fitness will be obtained for the line-segment sequence lengths (f_L), directions (f_D) and finish quadrants (f_{FQ}), together with a fitness (f_{SFQ}) for the finish quadrant for the whole sequence.

The chromosome fitness (F_c) was then calculated as the average of the above values (where $n = 2$),

$$f_L = \frac{1}{n}\sum_{i=1}^{n} f_L(i) \tag{5.6}$$

$$f_D = \frac{1}{n}\sum_{i=1}^{n} f_D(i) \tag{5.7}$$

$$f_{FQ} = \frac{1}{n}\sum_{i=1}^{n} f_{FQ}(i) \tag{5.8}$$

$$F_c = \frac{1}{4}\left(f_L + f_D + f_{FQ} + f_{SFQ}\right) \tag{5.9}$$

The calculation of a total fitness value using the summation of individual components is further discussed in Chapter 7.

5.6 Genetic Algorithm Behaviour

So that the behaviour of the standard genetic algorithm can be monitored and analysed satisfactorily, a display is presented for each genetic algorithm Test. The display is available at the end of each generation and at the end of the Test. Referring to the legend diagram, Figure 5-12 and Figure 5-31: Test 1, we have the following descriptions of the layout of the display:

1. Top Row. Program Name; Date; time; mutation probability; crossover probability; number of feature standard deviation in the fitness calculations; binary (16) or integer (5) chromosome; and genetic algorithm type, replacement or Random.
2. Rectangular Window. The horizontal co-ordinate is indexed by the value of the 10 right-most bits (0 to1023) in the chromosome, i.e. line-segment list start search index and ignore length threshold (parameter genes 4 and 5). The vertical co-ordinate is indexed by the observation

genes (0 to 63) in the chromosome, i.e. start quadrant, clockwise or anti-clockwise and the number of line-segments (observation genes 1, 2 and 3. The dots represent a chromosome. The crosses identify the chromosomes that have fitness values greater than a specified fitness threshold. The value of the fitness threshold is shown at the top right corner of the rectangular window. Note that only data from these chromosomes is recorded at the end of each generation for use by the recognition process (Chapter 6), thus reducing the amount of data to be analysed. The number of chromosomes in the population is shown at the bottom left corner and the generation number is shown at the bottom right corner of the rectangular window.

3. Fitness Histogram. This fitness histogram is scaled between 0.0 and 1.0, with fitness values less than zero being limited to zero. Ten bins are provided each with a width of 0.1. The histogram only shows the values for the current generation. This histogram is used for the fitness dispersion calculations (Lis, 1995).
4. Average Fitness Graph. This display shows the values of the population average fitness, where the two horizontal lines indicate the fitness values of 0.5 and 1.0 respectively. The two numbers, to the right of the graph, are the maximum fitness obtained up to the current generation (upper number) and the average fitness (lower number) for the current generation.
5. Unique Solutions Graph. This display shows the number of unique (individual) chromosomes that have been evolved up to the generation on display. The data is obtained from a 2D histogram array (indexed as 0 to 63, 0 to 1023) that is updated each generation with the information from those chromosomes that have a fitness greater than the fitness threshold. The count (displayed at the upper right of the graph) shows how many uniquely valued (individual) chromosomes have evolved. This histogram is recorded for use by the recognition process (see Chapter 6) in order to identify the unique solutions in the chromosome data recorded onto disc. A unique (individual) solution has a unique bit pattern in the chromosome, i.e. a unique number in the range 0 to 65535.
6. Diversity Graph. A measure of the diversity of the chromosomes in the population is

investigated in this chapter (see below for more details) and is displayed at the end of each generation in this graph. The vertical scale of the graph is logarithmic and the two horizontal lines show the approximate values 2.0 and 7.0 respectively. The two values at the bottom right of the display show the minimum diversity value (upper number) and the diversity value (lower number) for the current generation. Note that a low diversity value indicates a highly diverse set of chromosomes in the population. High diversity values indicate very little diversity in the population of chromosomes. Diversity measurements are discussed in section5.7.

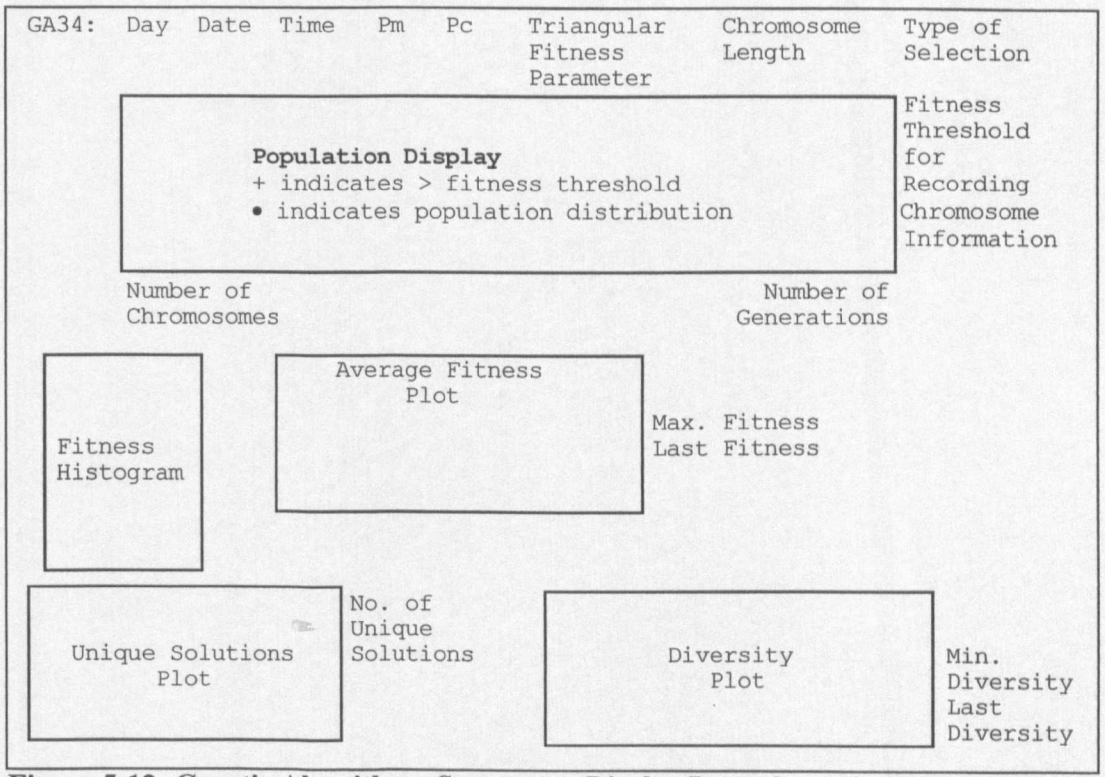

Figure 5-12: Genetic Algorithm - Summary : Display Legend

The displays Figure 5-31: Test 1 to Figure 5-74: Test 44 and the genetic algorithm test summary (see Figure 5-30) are used in the following sections to discuss and analyse the behaviour of the genetic algorithm for various values of the crossover and mutation probabilities. Particular attention is given to analysing the behaviour of the genetic algorithm that evolves a variety of solutions rather

than just the 'best' solution. The unique solutions column in Figure 5-30 identifies the number of unique solutions that are evolved more than once by the number in the square brackets, e.g. Test 1 30 [24]. Hence solutions appear and disappear and can return more than once (see also Chapter 8).

5.7 Dispersion and Diversity Measurements

Lis (1995) discussed a dispersion or diversity measure, which is based on the standard deviation of the fitness of the fittest chromosomes in the population. "If the mutation probability is established at a level that is too low, premature convergence occurs and a local minimum is usually identified. The histogram of the fitness function for the population takes the shape of a clear and sharp maximum, about which the majority of chromosomes are concentrated. On the other hand, when the mutation probability is too high the population is randomised and the histogram of the fitness function for the population will be of a more uniform shape with an even spread of fitness throughout the population. The selection process is then not able to push chromosomes up towards the higher values of fitness. When the mutation probability is established at the appropriate value, the histogram of the fitness values will show a peak close to the currently encountered best solution." However, the width of this peak is such that "the remaining body of the population is not neglected and is scattered outside this area of local maximum. These chromosomes search for possible new paths that may lead them to reaching a better maximum, possibly even a global one".

5.7.1 Dispersion Measurements

The method, proposed by Lis (1995), evaluated the concentration of the fitness function value close to the maximum fitness value. The fitness concentration measure, called the population fitness dispersion for generation i, $\rho(i)$, is calculated as follows:

$$\rho(i) = \frac{\sigma(n)}{\max(f)} \tag{5.10}$$

where $\sigma(n)$ is the fitness standard deviation for the fittest n chromosomes and $\max(f)$ is the fitness of the fittest chromosome in generation i. The mean population dispersion for generation i is averaged over the previous k generations and is given by the following calculation:

$$\rho_k(i) = \frac{1}{k} \sum_{j=i-k}^{i} \rho(j) \tag{5.11}$$

Lis (1995) included the top half $\left(n = \frac{N}{2}\right)$ of the chromosome population (N) in the calculation. The mutation probability was adjusted every five generations, so that the average fitness standard deviation for the previous twenty generations is controlled to a value between σ_{min} and σ_{max}. The range $\sigma_{max} - \sigma_{min}$ is reduced as the evolution progresses. Experimental results, from this paper, showed that the performance of the genetic algorithm improved with this type of dynamic adjustment to the mutation probability.

The research work has investigated the above fitness dispersion method of Lis (1995) to find out if this method can indicate the crossover and mutation probability required for the evolution of a number of solutions. The standard deviation of the chromosome fitness for the top 20% of the population was calculated and averaged over the previous five generations. Fitness dispersion versus generation-number graphs, Figure 5-13 to Figure 5-23, are shown for various values of the mutation probability, ranging from 0.01 to 1.0 with the crossover probability set to 0.6. For low values of the mutation probability, the fitness dispersion reached zero as the generations progress indicating that the fittest 20% of the population have the same fitness. For higher values of the mutation probability, the fitness dispersion failed to reach zero and for mutation probabilities greater than 0.2, the fitness dispersion maintained values between 0.15 and 0.2. This fixed value for the range of the fitness dispersion indicated that the diversity of the chromosome population was remaining approximately constant. Hence this form of dispersion or diversity measure, i.e. when the

fitness dispersion maintains values within a fixed band, could possibly be used to find the range of mutation probabilities that give a variety of solutions from the population of chromosomes.

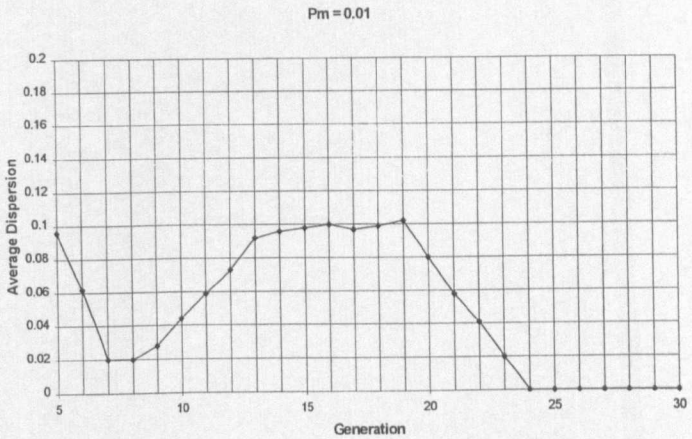

Figure 5-13: Fitness Dispersion for Pm = 0.01 and Pc = 0.6

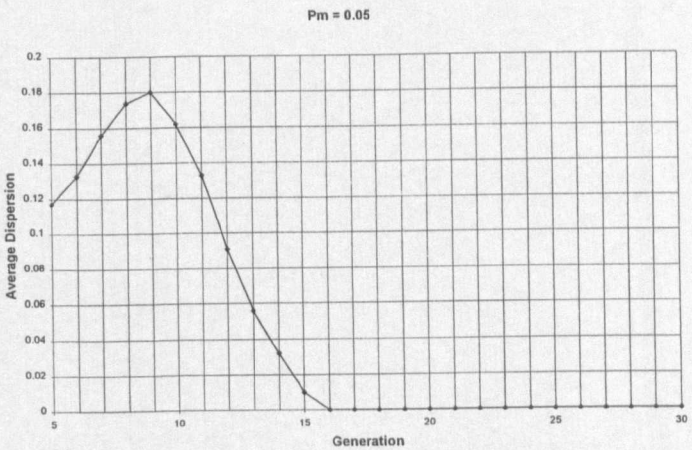

Figure 5-14: Fitness Dispersion for Pm = 0.05 and Pc = 0.6

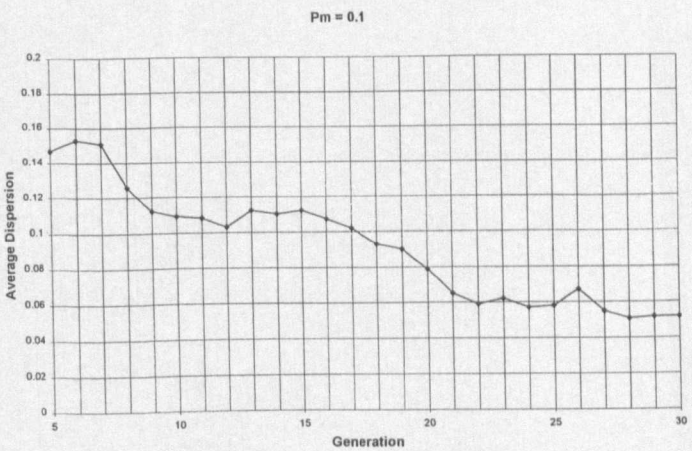

Figure 5-15: Fitness Dispersion for Pm = 0.1 and Pc = 0.6

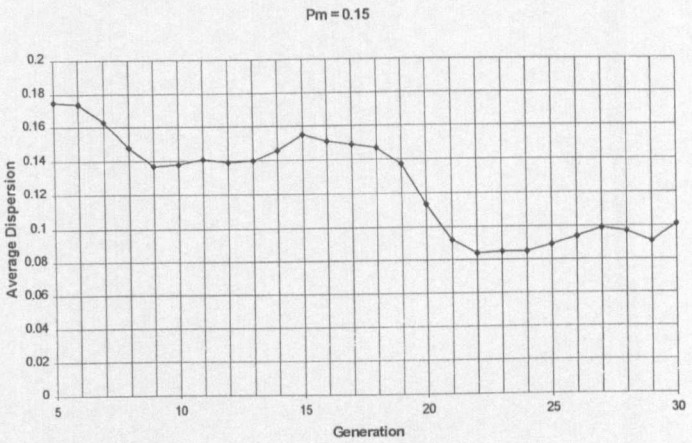

Figure 5-16: Fitness Dispersion for Pm = 0.15 and Pc = 0.6

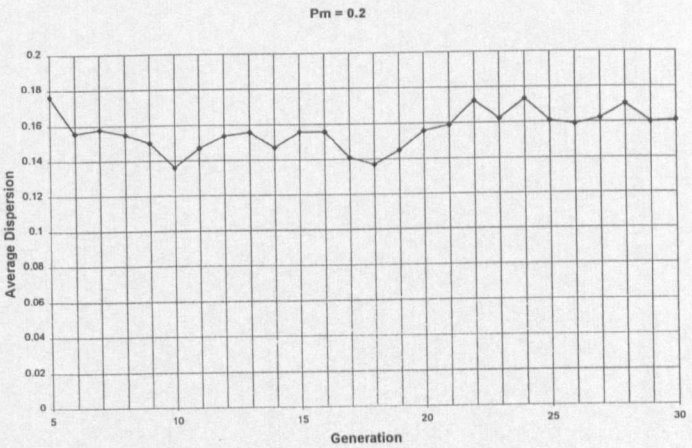

Figure 5-17: Fitness Dispersion for Pm = 0.2 and Pc = 0.6

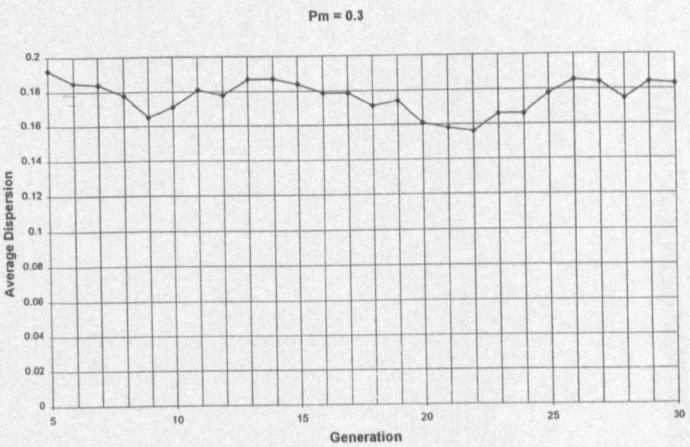

Figure 5-18: Fitness Dispersion for Pm = 0.3 and Pc = 0.6

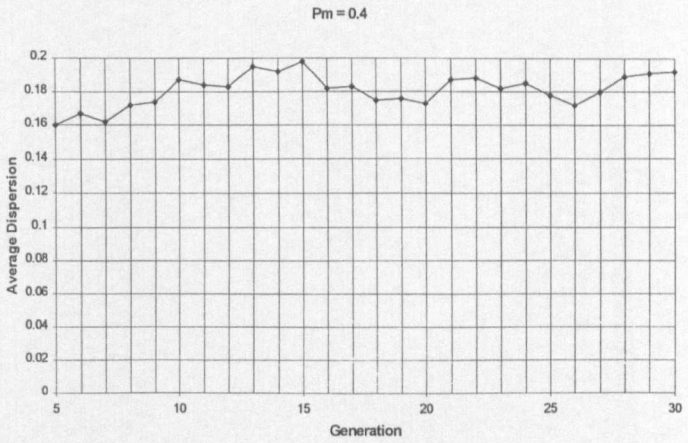

Figure 5-19: Fitness Dispersion for Pm = 0. 4 and Pc = 0.6

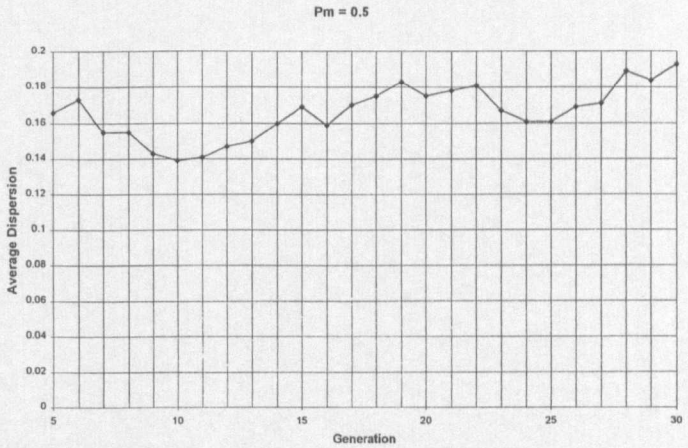

Figure 5-20: Fitness Dispersion for Pm = 0. 5 and Pc = 0.6

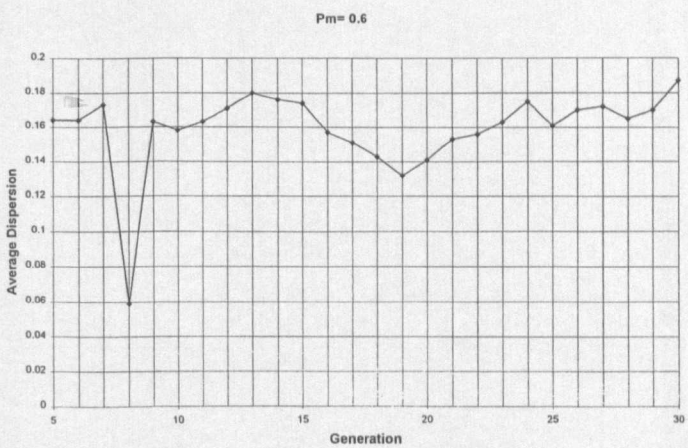

Figure 5-21: Fitness Dispersion for Pm = 0. 6 and Pc = 0.6

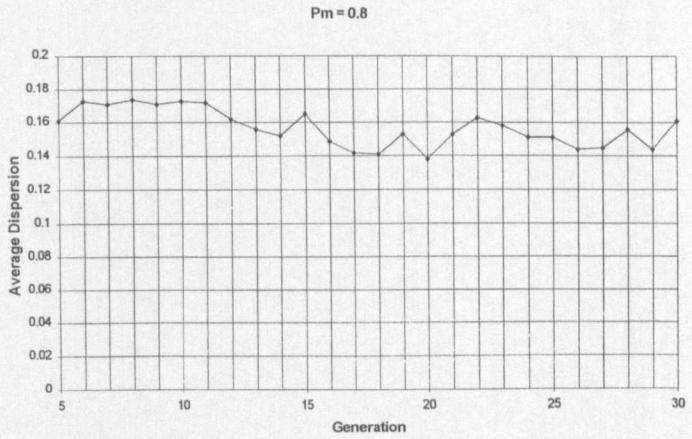

Figure 5-22: Fitness Dispersion for Pm = 0.8 and Pc = 0.6

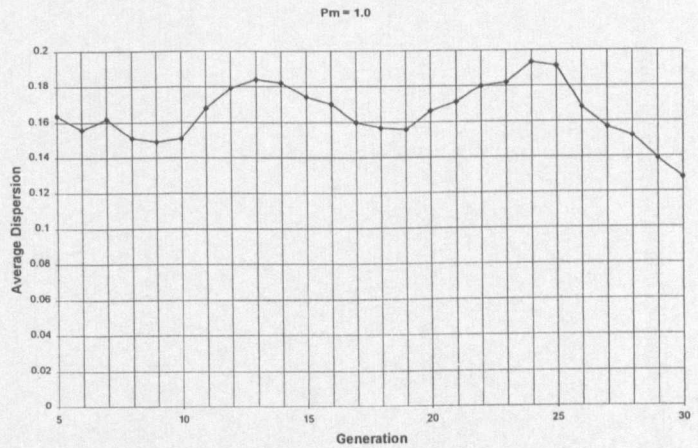

Figure 5-23: Fitness Dispersion for Pm = 1.0 and Pc = 0.6

5.7.2 Diversity Measurements

The following new diversity measure has been developed to investigate how to set the value of the crossover probability and mutation probabilities in order to obtain a selection of solutions rather than just the 'best' solution. The research experiments described below were conducted with the crossover probability equal to 0.6. A mutation probability that is too low may result in premature convergence, due to the strong influence of the crossover operator. A mutation probability that is too high often produces an approximately random search process, due to the predominance of the mutation operator over the crossover operator. The genetic algorithm literature describes various methods for setting the crossover and mutation probabilities in order to find the 'best' solution. As the 'best' solution evolves, the average fitness of the population increases as more chromosomes develop high fitness values. Eventually a large proportion of the population has a similar fitness to

the 'best' chromosome. Thus the population in this case can be said to have 'low diversity'. Under these circumstances each chromosome has a similar bit pattern. When a number of solutions are required, the crossover and mutation probabilities have to be adjusted to increase the diversity of the chromosomes such that a variety of solutions are evolved, with lower fitness, near to the 'best' solution.

The genetic algorithm described in this chapter was required to evolve as many different ways of 'looking at' or observing a contour as possible, which have fitness values near to the fitness of the 'best' way to look at the contour. Therefore, a chromosome diversity measure calculated from the 'ways of looking' at a contour, i.e. the observation genes in the chromosome, could possibly be used more directly to monitor the effect of the mutation probability on the evolution of a variety of solutions. This new type of diversity measure was calculated as follows:

1. A histogram is collected (diversity histogram) using the observation genes of the chromosome as the bin index. The bin index has values ranging from 0 to 63, i.e. 16 x (gene 1) + 8 x (gene 2) + (gene 3). This diversity histogram identified how many different ways of observing the contour are evolved. Each non-zero bin index decoded into a particular way in which to look at the contour.
2. The mean value for the contents of this diversity histogram was calculated. The total content will equal the population size. The mean diversity histogram value will, thus, equal the population size divided by 64, which for a population of 50 will equal 0.7813.
3. The standard deviation for the diversity histogram was calculated. The maximum value for this standard deviation will be when all the chromosomes have the same observation genes' pattern and is equal to 6.2010 for a population of 50 chromosomes. The minimum value for this standard deviation will be when each bin has the value one, which for a population of 50 chromosomes will have the value 0.4134.
4. The chromosome diversity was defined as the diversity histogram standard deviation divided by

the diversity histogram mean. The maximum and minimum values for this population diversity measure are 7.9373 and 0.5292 respectively. Note that this population diversity measure included the whole population and only part of the chromosome, whereas the fitness dispersion of Lis (1995) only considered the fitness of the fittest 20% of the population.

High chromosome diversity was indicated by a low diversity value and low chromosome diversity is shown by a large diversity value. This new type of diversity measure indicated how well the genetic algorithm was performing when evolving a number of solutions rather than just the 'best' solution. Figure 5-24 to Figure 5-29 show the variation in chromosome diversity for a variety of crossover and mutation probabilities. In a similar manner to the fitness dispersion method of Lis (1995), the new diversity measure decreased to values in the range 0.0 to 2.0 as the mutation probability was increased. This range of diversity values was obtained for mutation probabilities greater than 0.2.

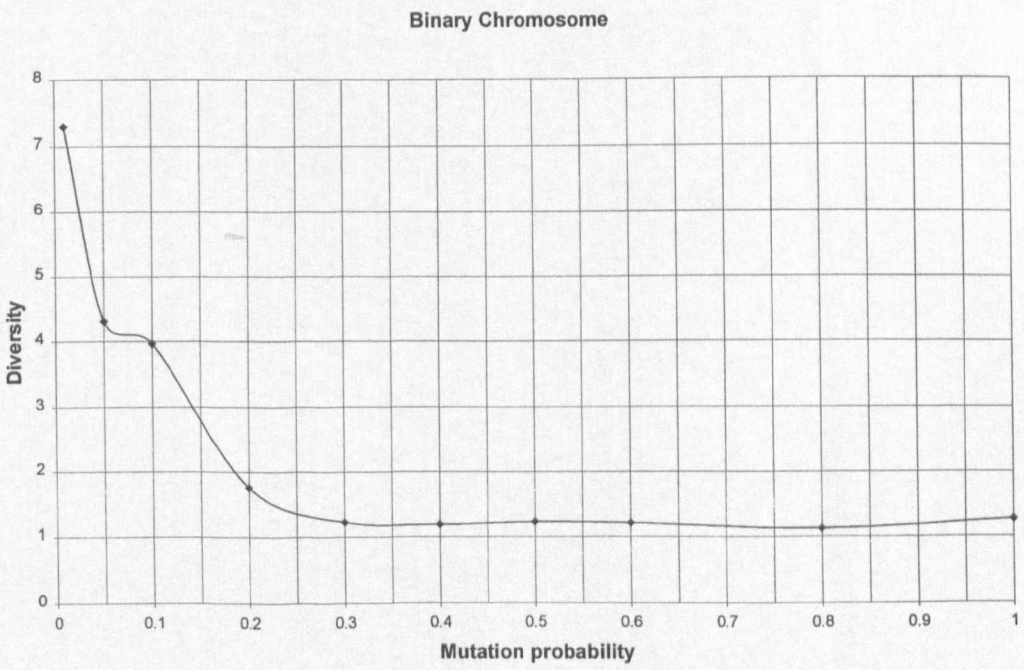

Figure 5-24: Diversity Values for Pc = 0.6 with Binary Chromosome

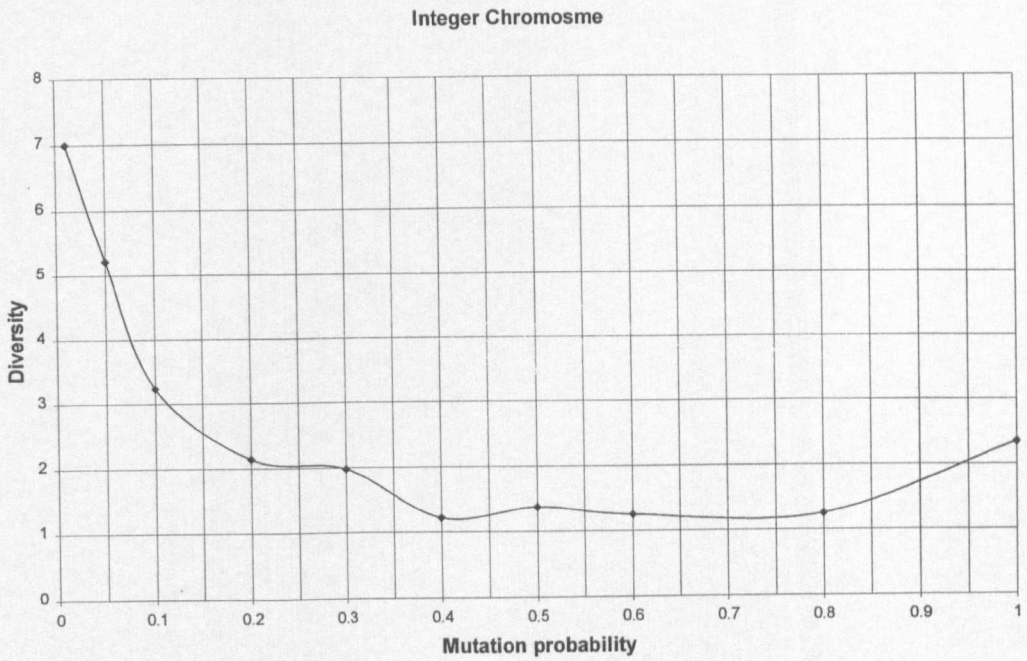

Figure 5-25: Diversity Values for Pc = 0.6 with Integer Chromosome

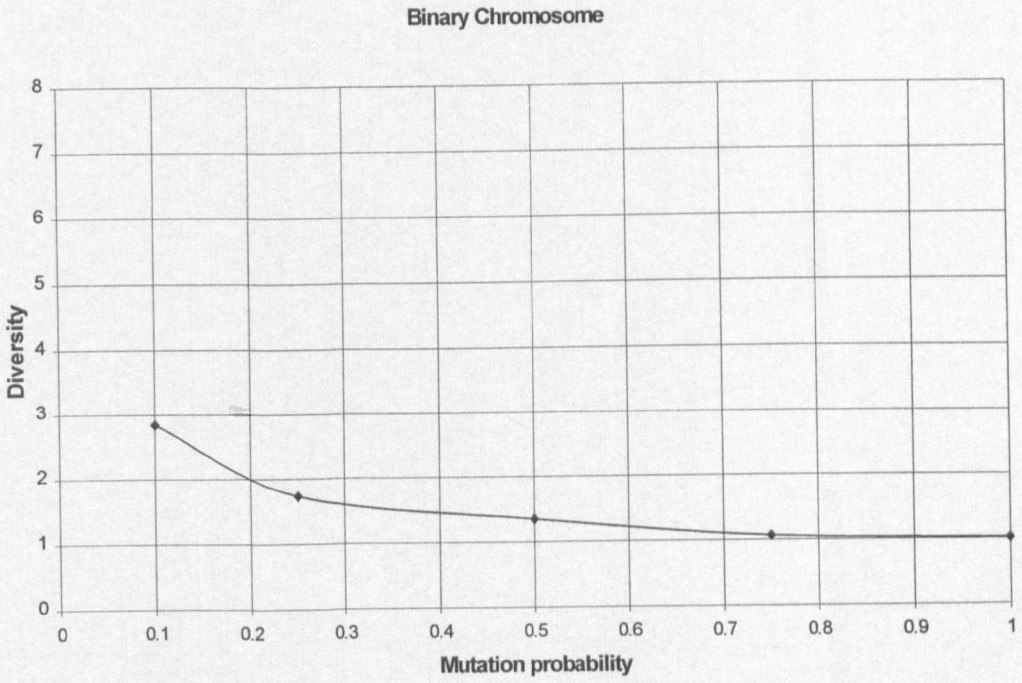

Figure 5-26: Diversity Values for Pc = 0.0 with Binary Chromosome

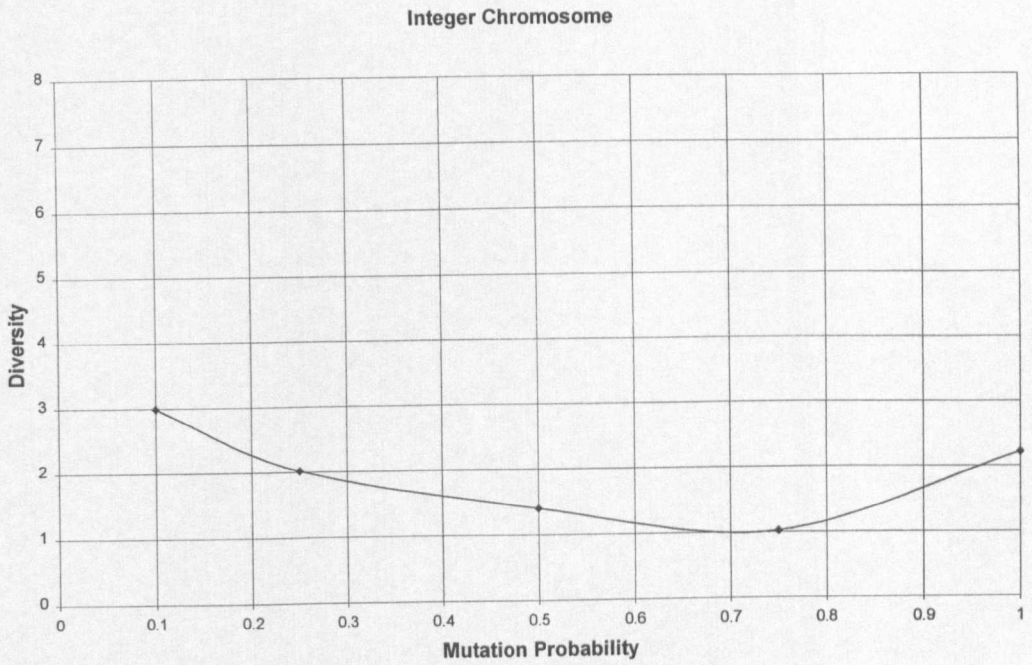

Figure 5-27: Diversity Values for Pc = 0.0 with Integer Chromosome

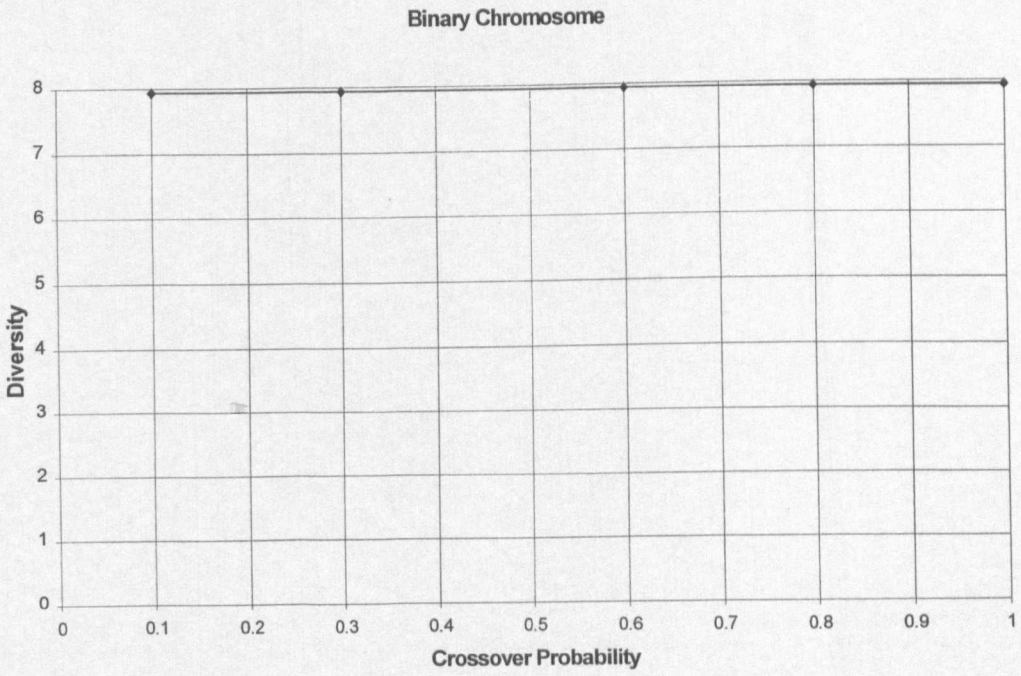

Figure 5-28: Diversity Values for Pm = 0.0 with Binary Chromosome

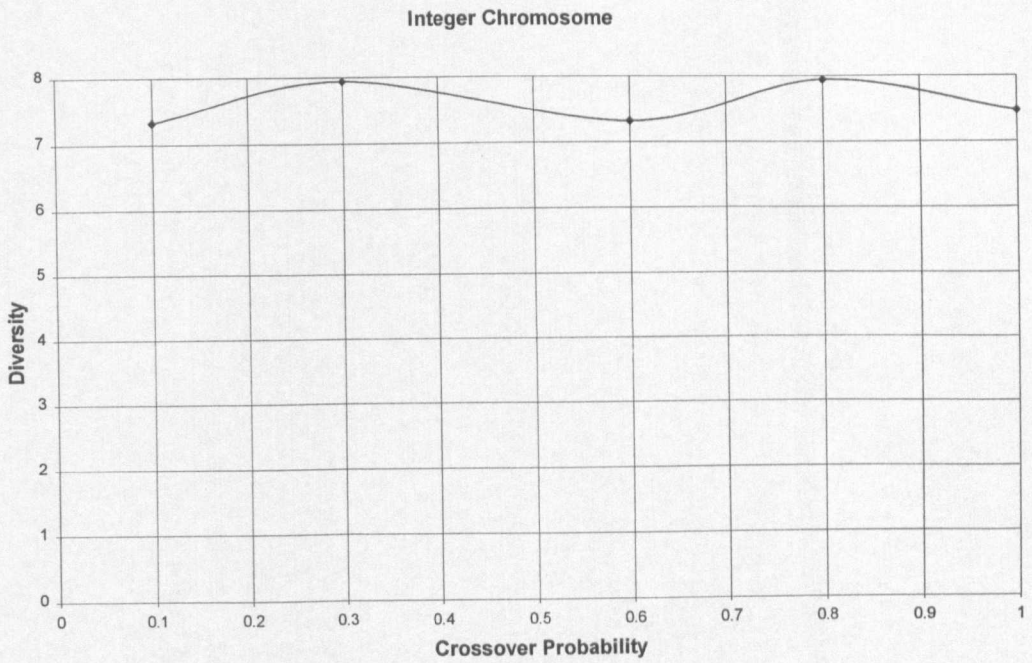

Figure 5-29: Diversity Values for Pm = 0.0 with Integer Chromosome

The results from these two methods appeared to be consistent. The new diversity histogram method may have advantages because it appears to have a sharper 'cut-off' at mutation probability values around 0.2 than the fitness dispersion method of Lis (1995). A further advantage may also occur because the new diversity histogram is describing, in a more direct manner, the diversity in the ways of 'looking at' or observing a contour. The dispersion in the chromosome fitness, on the other hand, is a more indirect measure of the chromosome diversity, because fitness is used primarily for the selection process by the genetic algorithm and its distribution may not reflect the true diversity of the genes in the chromosome population.

5.8 Genetic Operator Experiments and Results

The experimental results from the research work on genetic operator variations are shown in Figure 5-31: Test 1 to Figure 5-74: Test 44. These figures display the output from a selection of various experiments that have been performed using the genetic algorithm described in this chapter. (A legend diagram for these genetic algorithm displays is shown in Figure 5-12). Figure 5-30 summarises typical results for the forty-four tests.

Test	Figure Ident.	Chromosome Type	Pm	Pc	Unique Solutions	Different Solutions	Diversity
1	282042	Binary	0.01	0.6	30 [24]	2	7.30
2	291745	Integer	0.01	0.6	14 [12]	3	7.00
3	291932	Binary	0.05	0.6	39 [24]	6	4.30
4	291948	Integer	0.05	0.6	21 [18]	3	5.20
5	301715	Binary	0.10	0.6	53 [35]	6	3.90
6	301730	Integer	0.10	0.6	17 [14]	5	3.30
7	61603	Binary	0.20	0.6	38 [19]	5	1.76
8	61620	Integer	0.20	0.6	23 [14]	7	2.14
9	61633	Binary	0.30	0.6	28 [7]	4	1.23
10	61644	Integer	0.30	0.6	25 [13]	4	1.98
11	61657	Binary	0.40	0.6	17 [2]	3	1.19
12	61708	Integer	0.40	0.6	24 [3]	5	1.21
13	61742	Binary	0.50	0.6	11 [0]	4	1.23
14	61755	Integer	0.50	0.6	14 [4]	5	1.37
15	61807	Binary	0.60	0.6	7 [1]	5	1.21
16	61950	Integer	0.60	0.6	10 [1]	3	1.25
17	71720	Binary	0.80	0.6	9 [0]	6	1.12
18	71731	Integer	0.80	0.6	13 [4]	5	1.25
19	71745	Binary	1.00	0.6	5 [0]	3	1.27
20	71908	Integer	1.00	0.6	8 [4]	3	2.35
21	301744	Binary	N/A	N/A	7 [0]	5	1.23
22	301756	Integer	N/A	N/A	6 [1]	4	1.46
23	301943	Binary	0.10	0.0	26 [19]	3	2.84
24	301955	Integer	0.10	0.0	7 [5]	3	2.99
25	11830	Binary	0.25	0.0	17 [3]	6	1.73
26	11931	Integer	0.25	0.0	24 [16]	7	2.02
27	11950	Binary	0.50	0.0	6 [1]	2	1.35
28	12001	Integer	0.50	0.0	8 [5]	5	1.40
29	41514	Binary	0.75	0.0	8 [1]	5	1.07
30	42026	Integer	0.75	0.0	14 [7]	5	1.02
31	42039	Binary	1.00	0.0	8 [0]	4	1.02
32	51000	Integer	1.00	0.0	4 [2]	3	2.22
33	51138	Binary	0.00	0.1	3 [2]	1	7.94
34	51150	Integer	0.00	0.1	3 [2]	2	7.32
35	51203	Binary	0.00	0.3	0 [0]	0	7.94
36	51211	Integer	0.00	0.3	11 [11]	2	7.94
37	51416	Binary	0.00	0.6	21 [17]	1	7.94
38	51428	Integer	0.00	0.6	1 [1]	1	7.32
39	51441	Binary	0.00	0.8	12 [12]	3	7.94
40	51452	Integer	0.00	0.8	9 [9]	2	7.94
41	51542	Binary	0.00	1.0	17 [11]	3	7.94
42	51553	Integer	0.00	1.0	6 [5]	3	7.47
43	62028	Binary	0.50	0.6	12 [0]	8	1.10
44	62042	Integer	0.50	0.6	24 [0]	14	1.42

Figure 5-30: Genetic Algorithm Test Summary (Test 1 to Test 44)

Figure 5-31: Test 1 to Figure 5-50: Test 20 display the results for a crossover probability of 0.6 and a fitness threshold of 0.6 that was set to limit the amount of chromosome information recorded onto disc. Data for the binary and integer chromosomes is shown alternately. For mutation probabilities in the range 0.01 to 0.1 the average fitness increased with each generation, with the final average fitness very close to the maximum of 0.81. In this range of mutation probabilities, the genetic algorithm was evolving the 'best' solution in the manner of the standard genetic algorithm. The diversity measure was, in general, greater than 2.0, showing low diversity, i.e. most of the chromosomes in the population were be similar. For mutation probability values in the range 0.2 to 0.3, the average fitness value decreased to the values 0.3 to 0.4 and the diversity had values of about 2.0. The population chromosome diversity was beginning to increase with the corresponding decrease in the average fitness. The chromosomes now had a variety of fitness values, instead of evolving values around the maximum fitness value. For mutation probabilities greater than 0.3, the average fitness of the population decreased to values of about 0.2 and the diversity maintained values below 2.0. The population maintained similar average values and similar diversity levels throughout the 30 generations. The variation of chromosome diversity with mutation probability for these test cases is shown in Figure 5-24 and Figure 5-25.

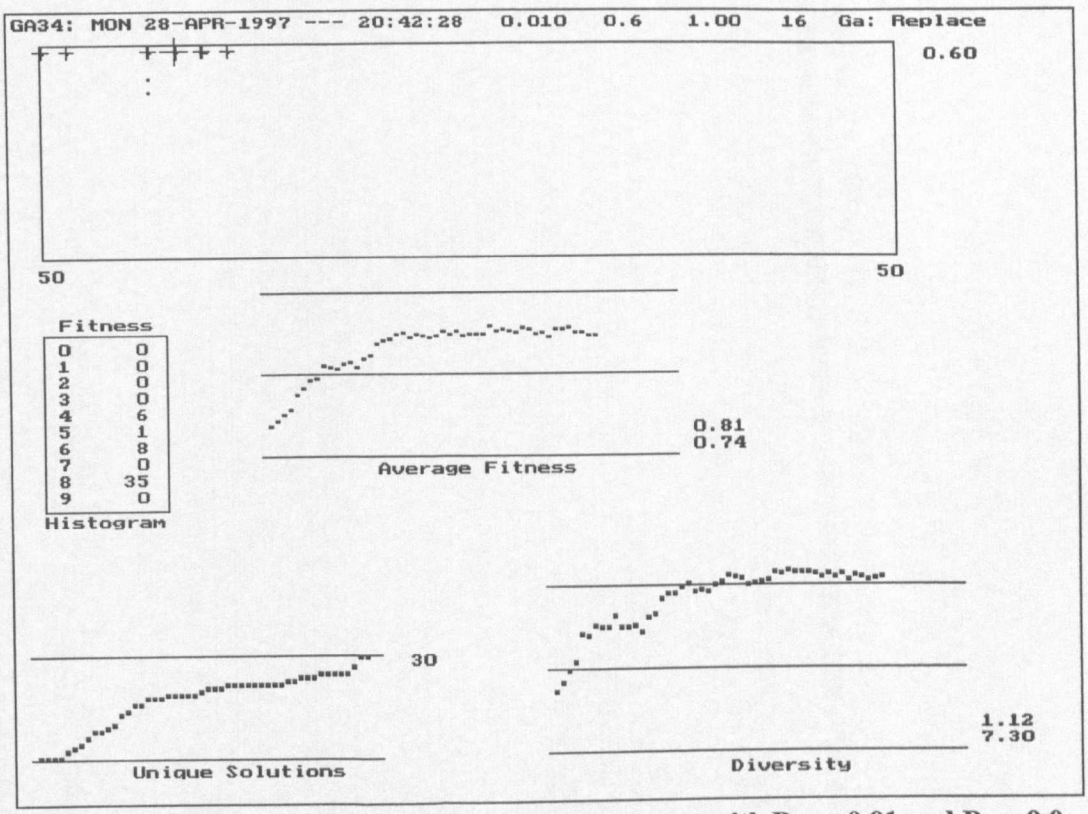

Figure 5-31: Test 1 GA Summary - Binary Chromosome with Pm = 0.01 and Pc = 0.0

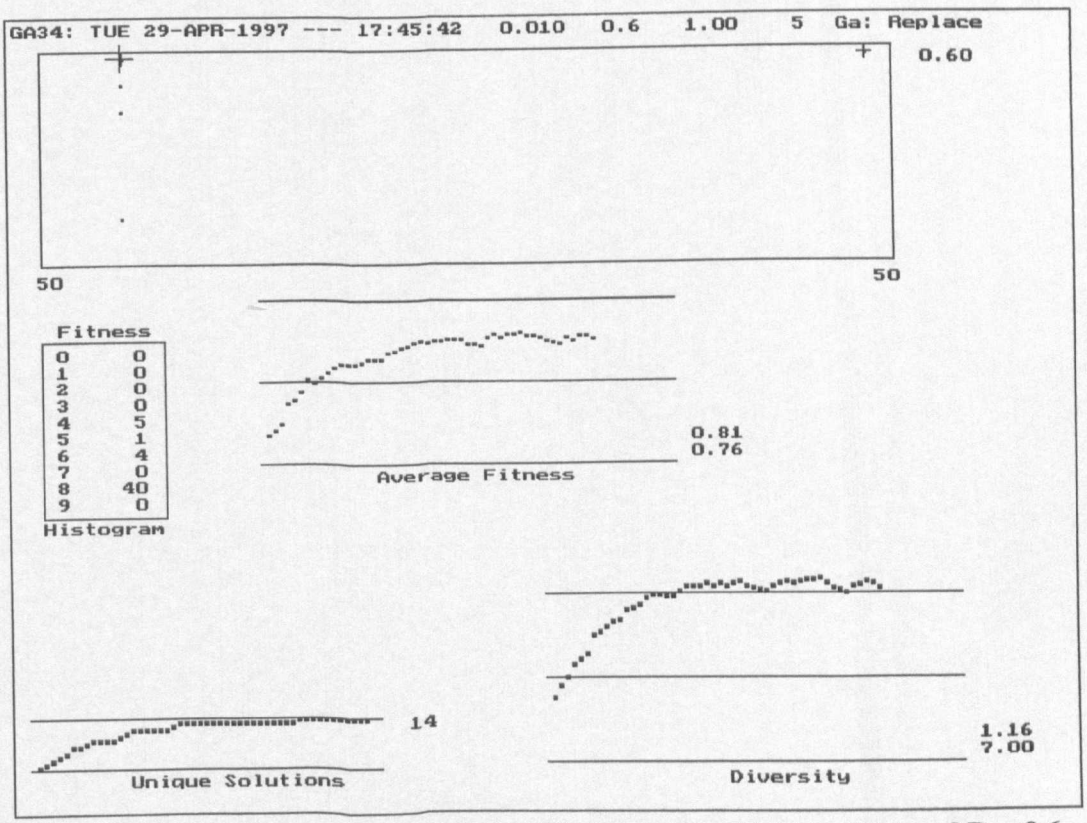

Figure 5-32: Test 2 GA Summary - Integer Chromosome with Pm = 0.01 and Pc =0.6

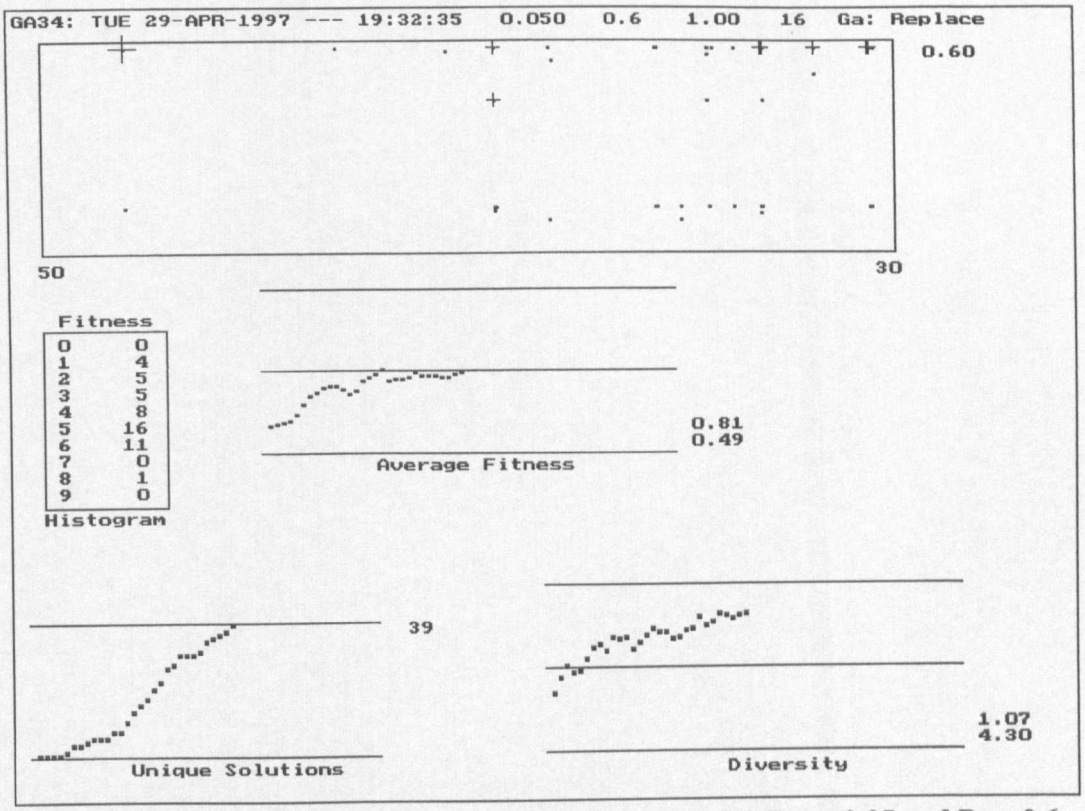

Figure 5-33: Test 3 GA Summary - Binary Chromosome with Pm = 0.05 and Pc = 0.6

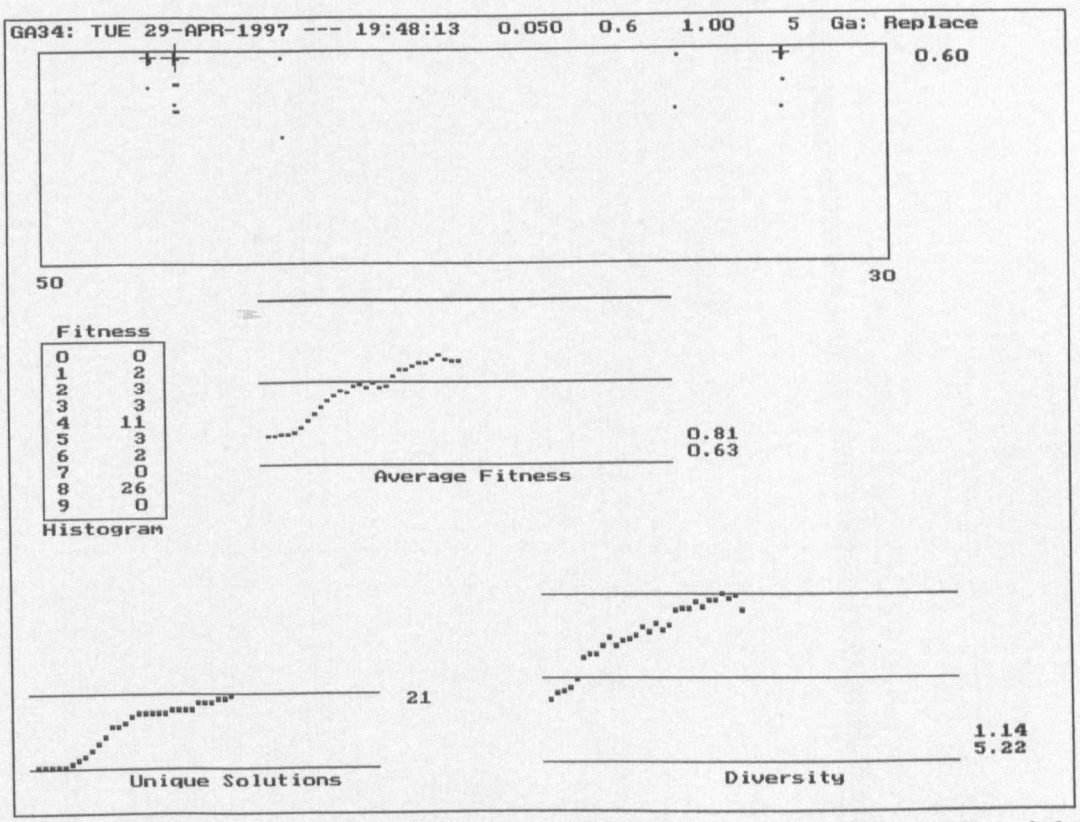

Figure 5-34: Test 4 GA Summary - Integer Chromosome with Pm = 0.05 and Pc = 0.6

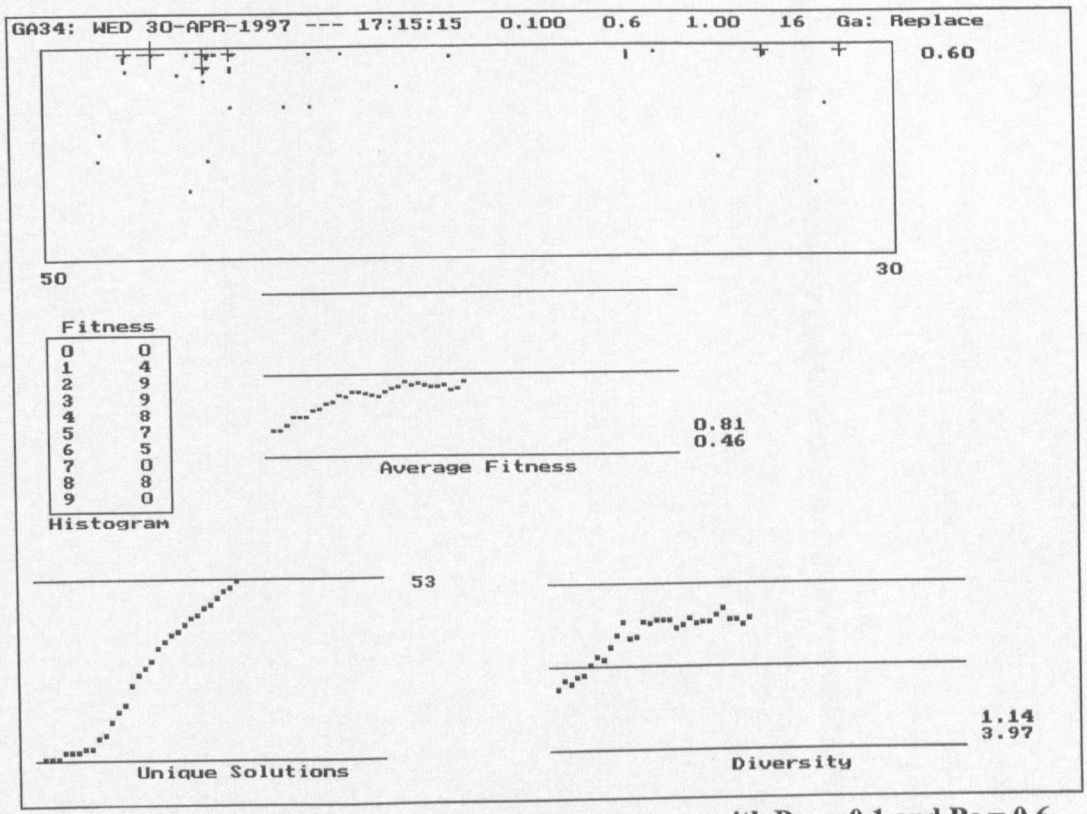

Figure 5-35: Test 5 GA Summary - Binary Chromosome with Pm = 0.1 and Pc = 0.6

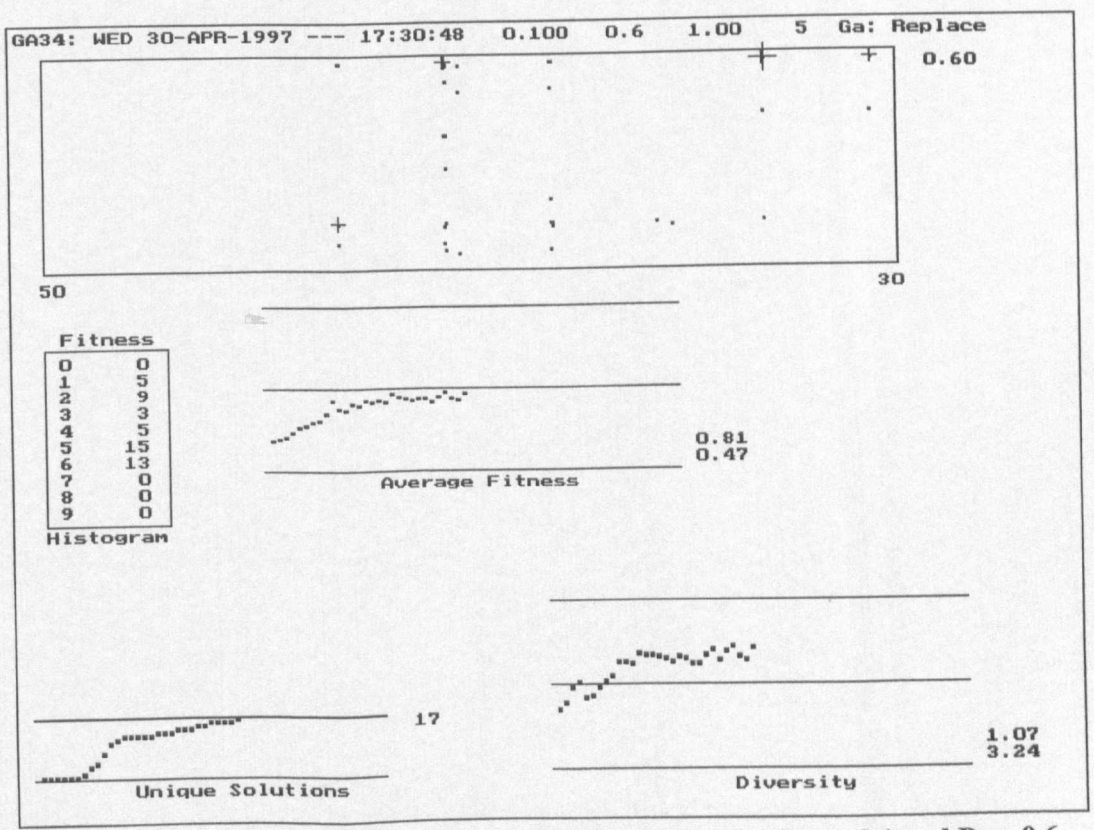

Figure 5-36: Test 6 GA Summary - Integer Chromosome with Pm = 0.1 and Pc = 0.6

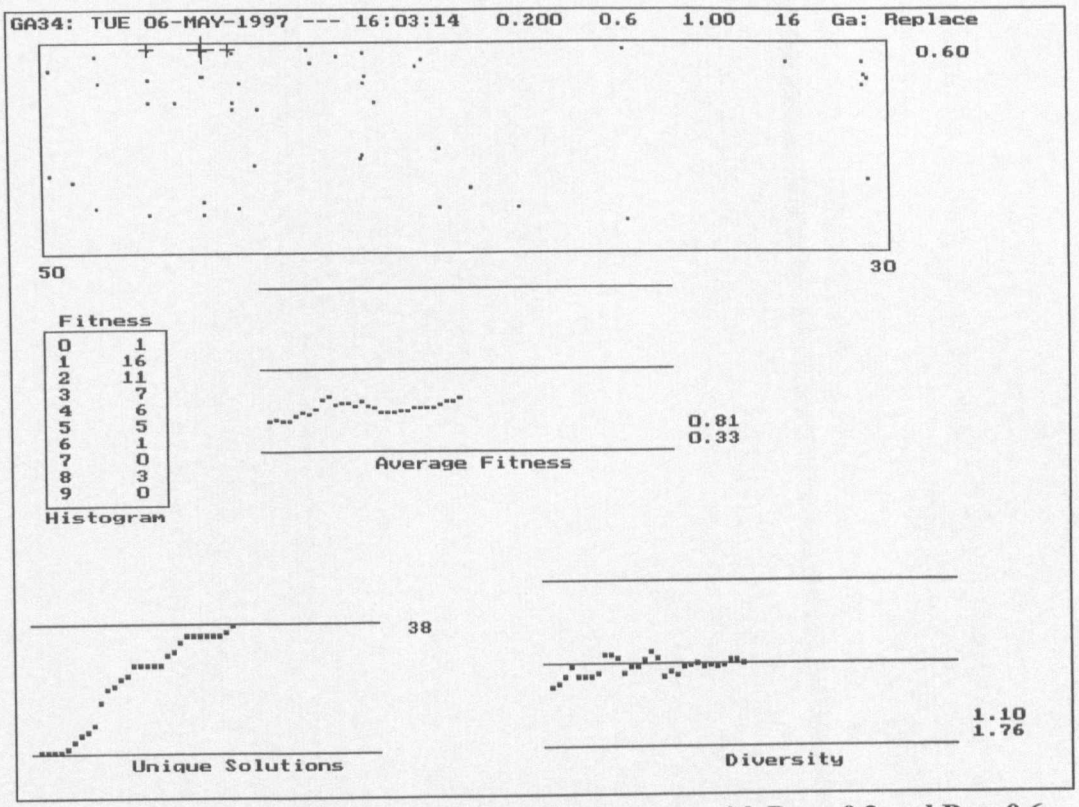

Figure 5-37: Test 7 GA Summary - Binary Chromosome with Pm = 0.2 and Pc = 0.6

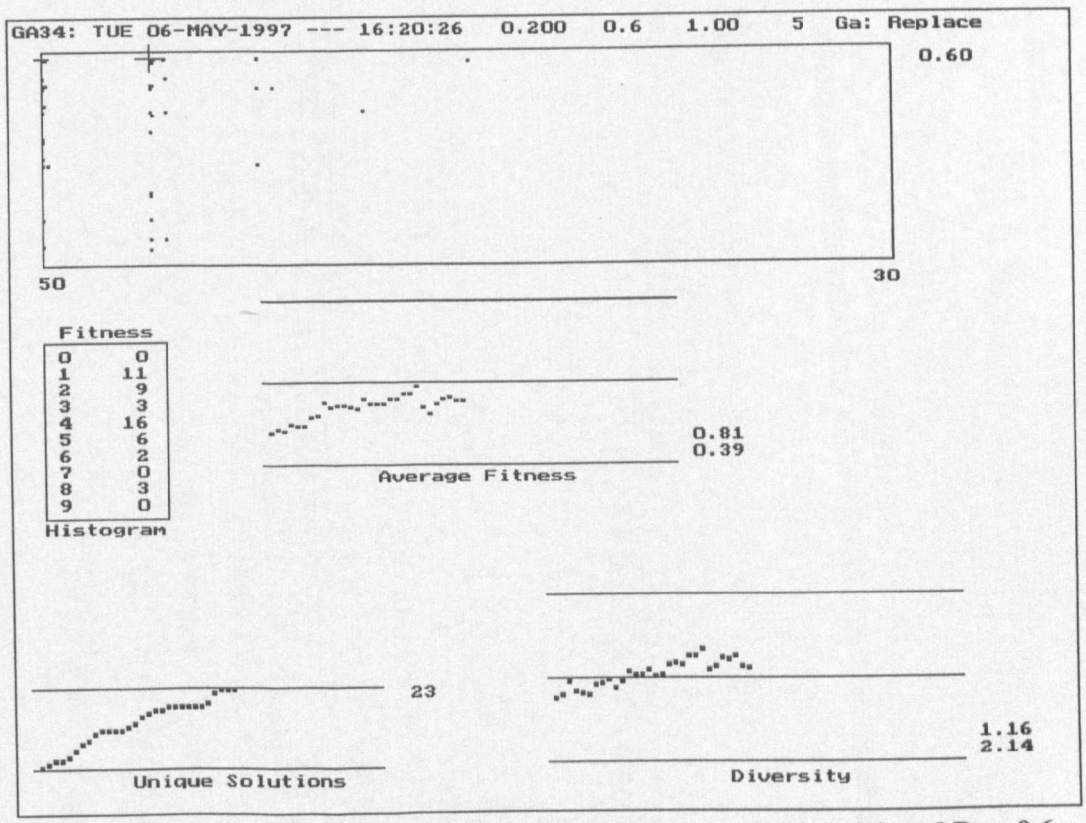

Figure 5-38: Test 8 GA Summary - Integer Chromosome with Pm = 0.2 and Pc = 0.6

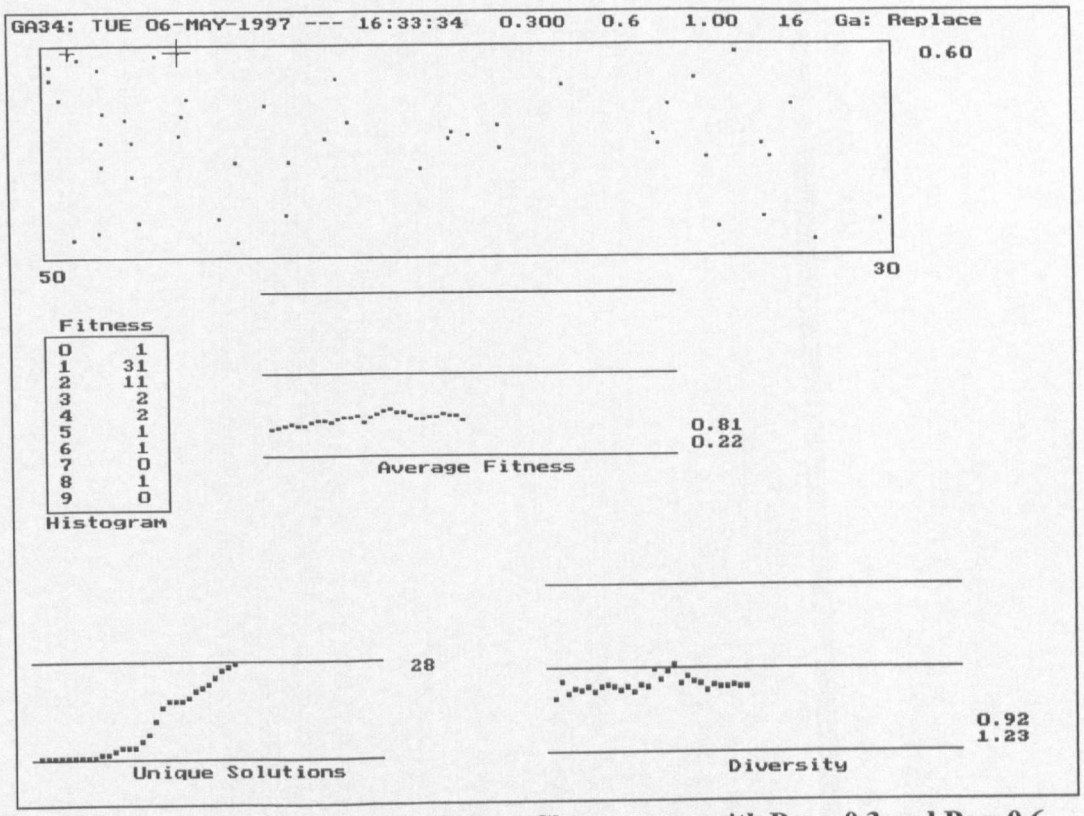

Figure 5-39: Test 9 GA Summary - Binary Chromosome with Pm = 0.3 and Pc = 0.6

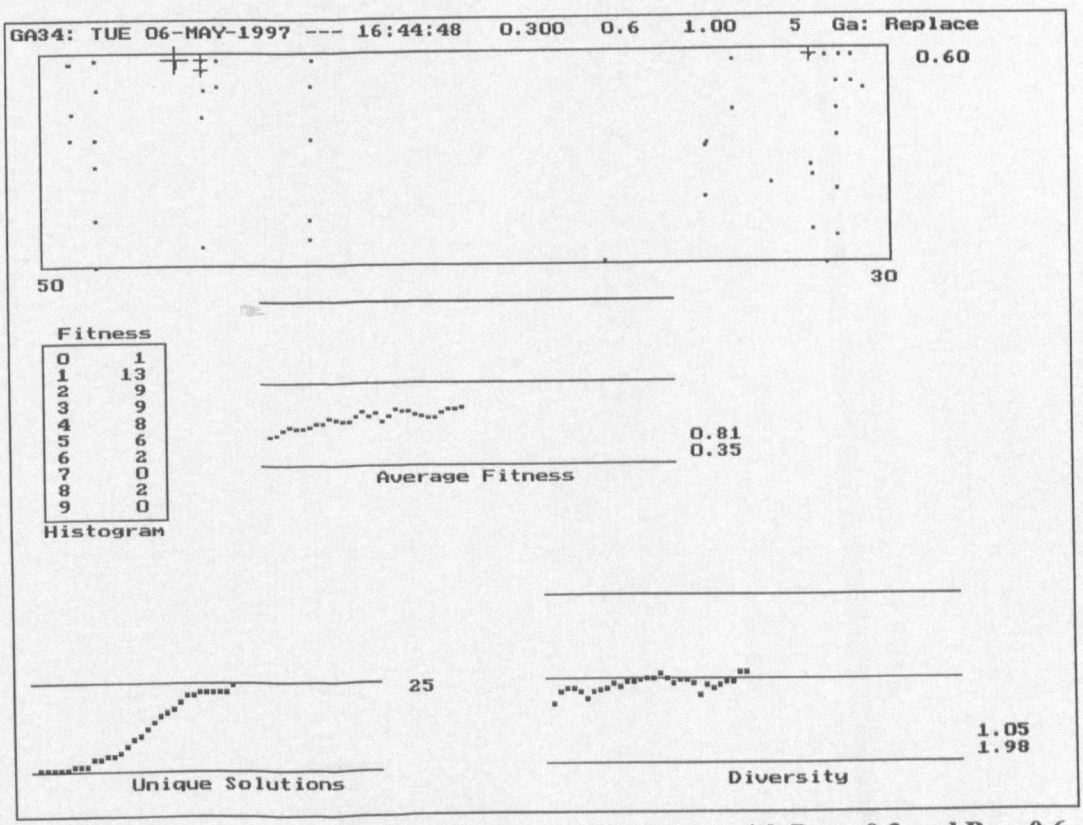

Figure 5-40: Test 10 GA Summary - Integer Chromosome with Pm = 0.3 and Pc = 0.6

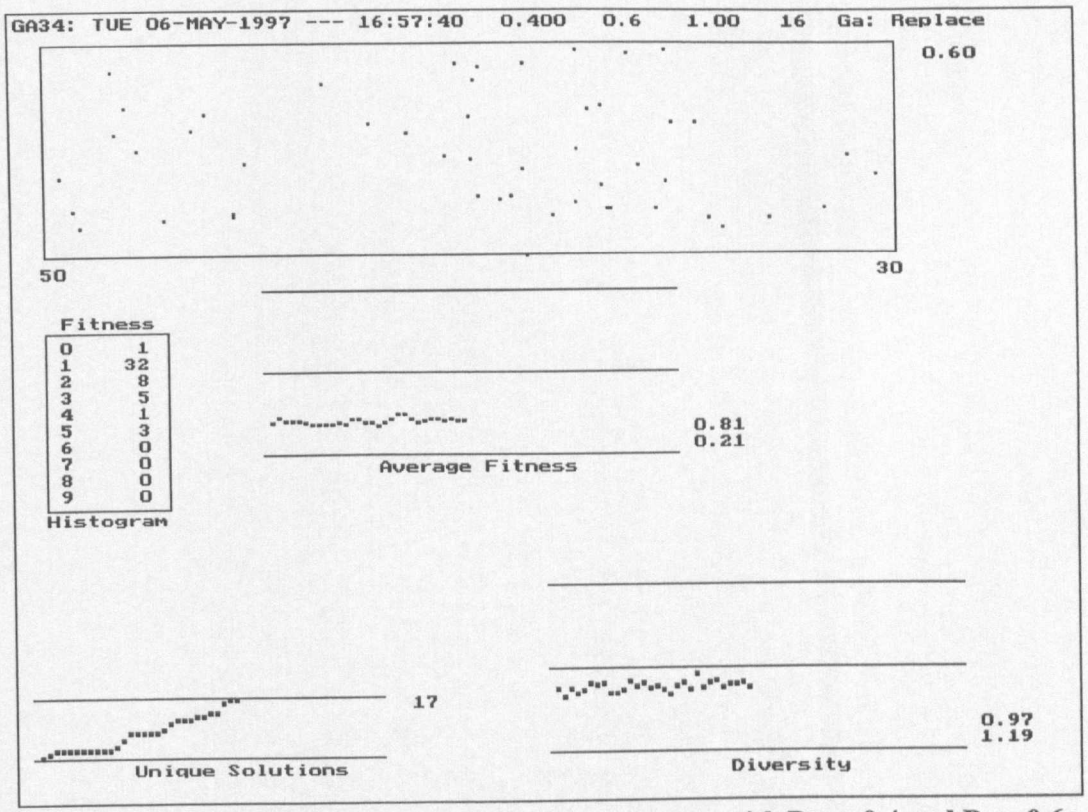

Figure 5-41: Test 11 GA Summary - Binary Chromosome with Pm = 0.4 and Pc = 0.6

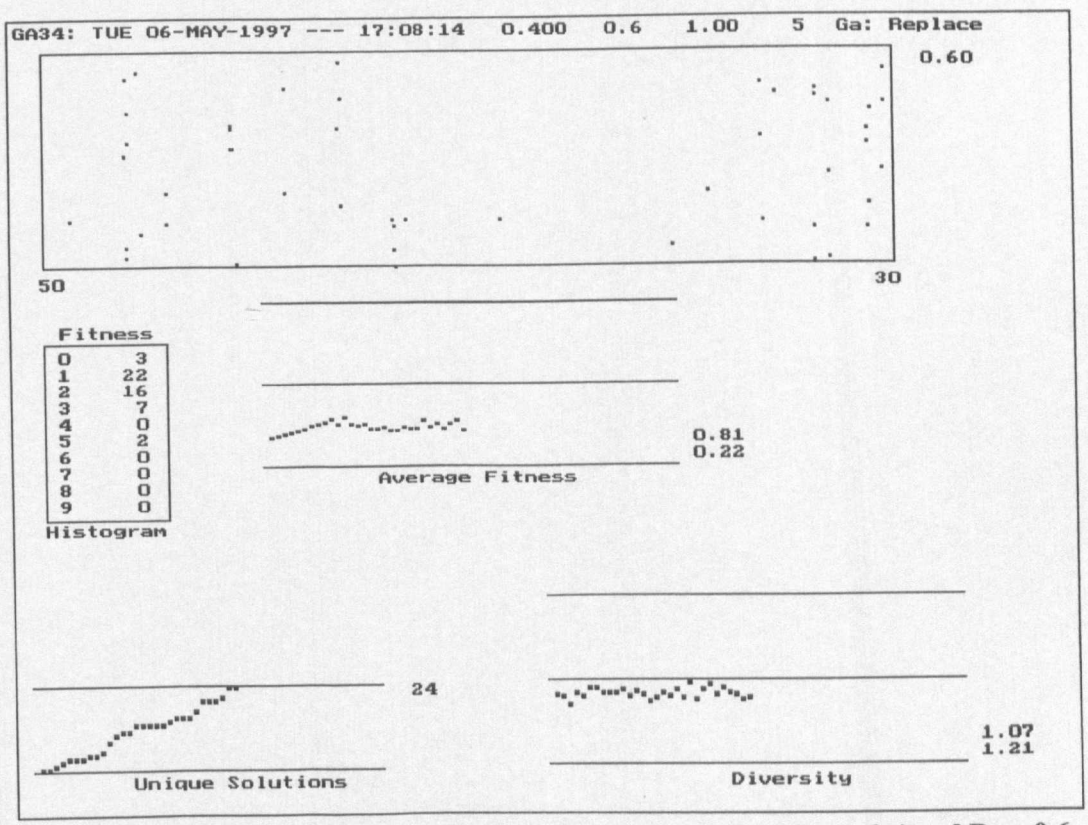

Figure 5-42: Test 12 GA Summary - Integer Chromosome with Pm = 0.4 and Pc = 0.6

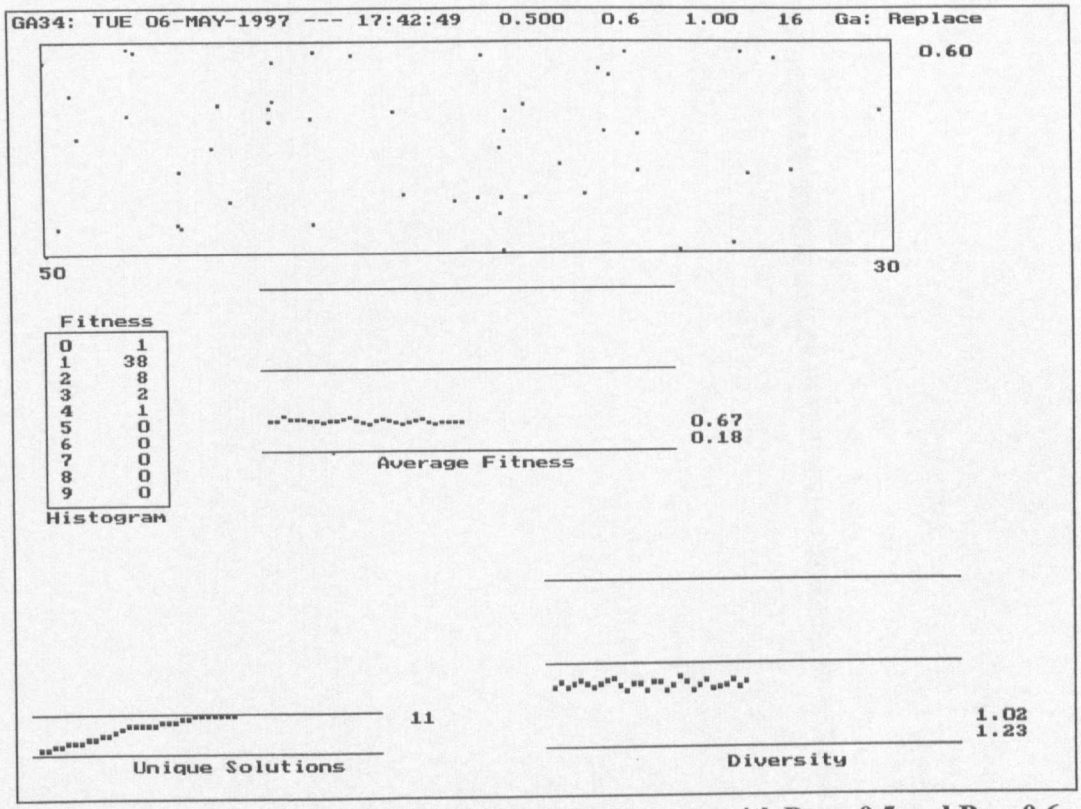

Figure 5-43: Test 13 GA Summary - Binary Chromosome with Pm = 0.5 and Pc = 0.6

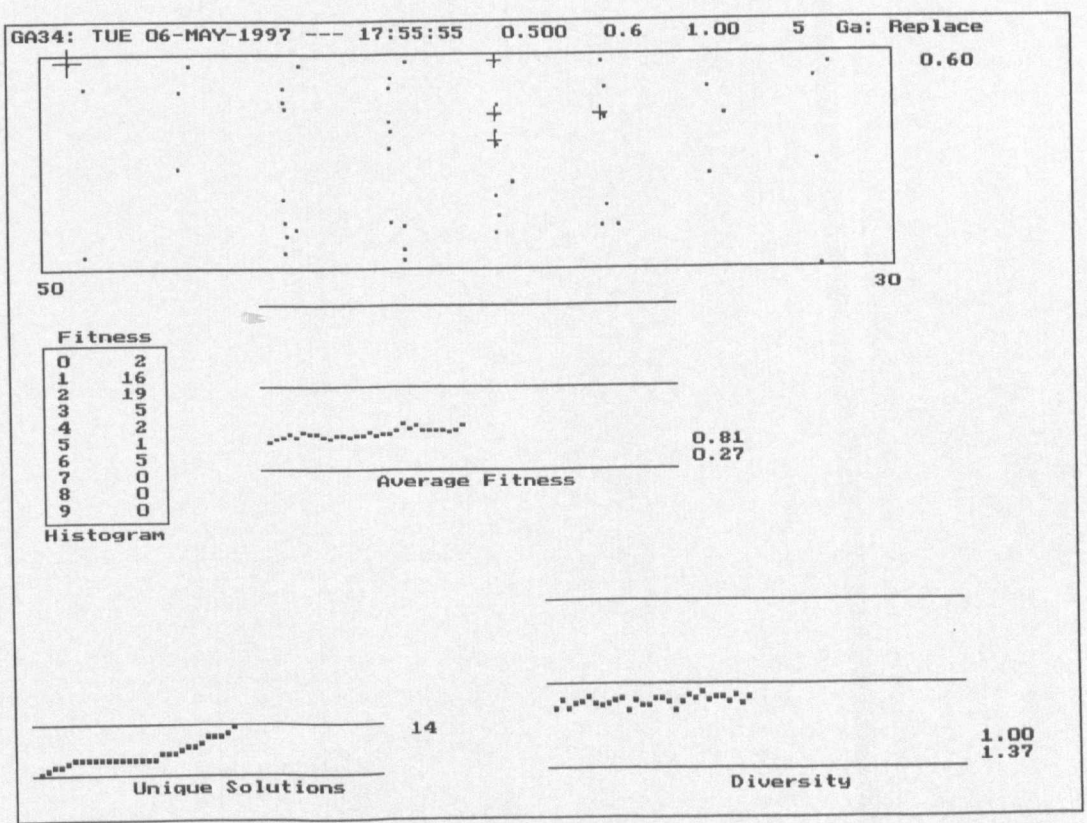

Figure 5-44: Test 14 GA Summary - Integer Chromosome with Pm = 0.5 and Pc = 0.6

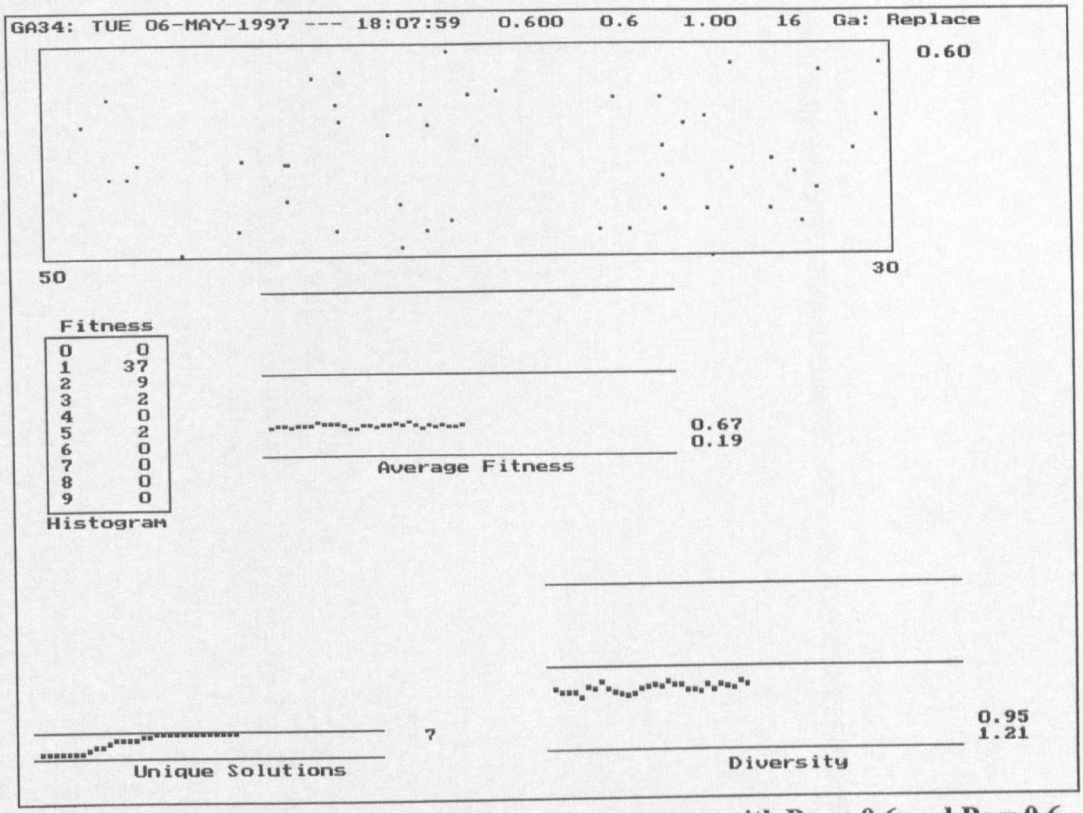

Figure 5-45: Test 15 GA Summary - Binary Chromosome with Pm = 0.6 and Pc = 0.6

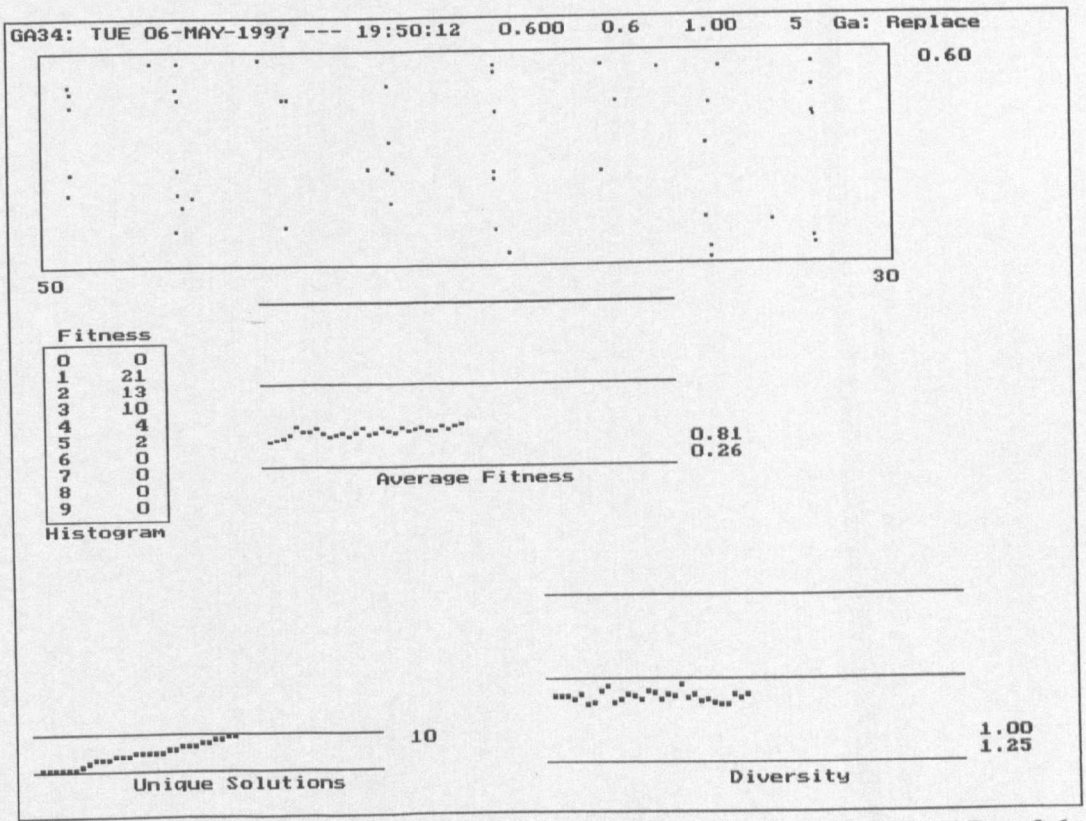

Figure 5-46: Test 16 GA Summary - Integer Chromosome with Pm = 0.6 and Pc = 0.6

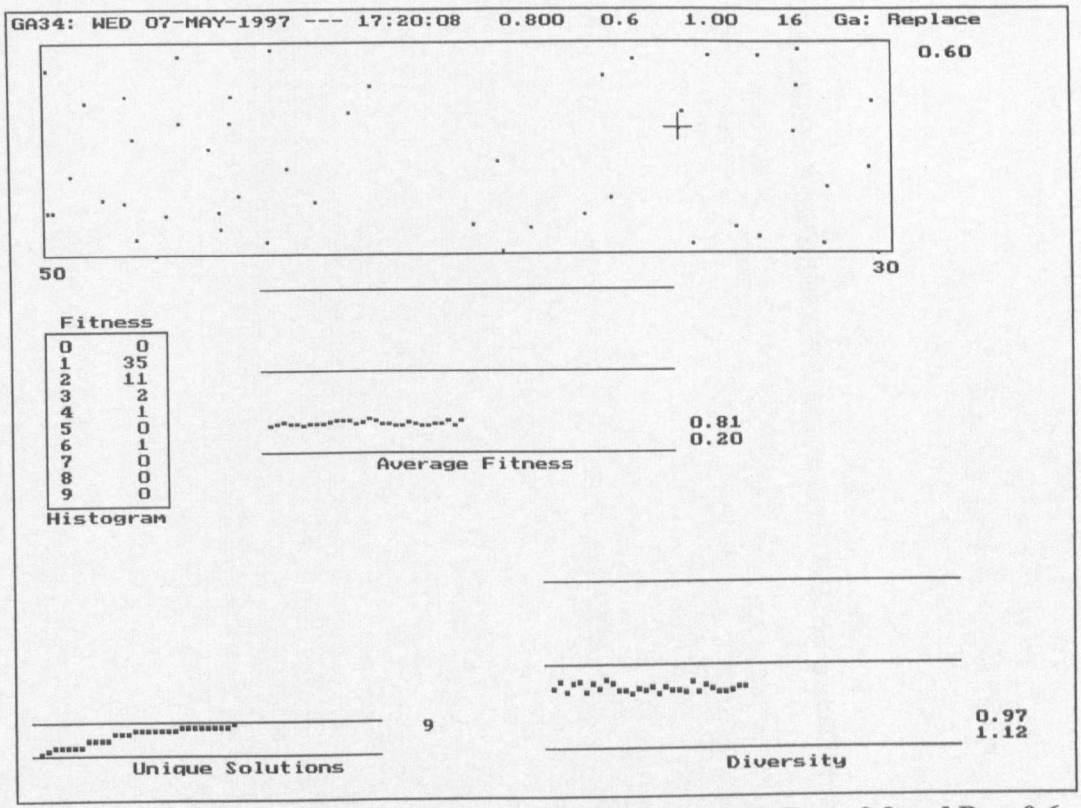

Figure 5-47: Test 17 GA Summary - Binary Chromosome with Pm = 0.8 and Pc = 0.6

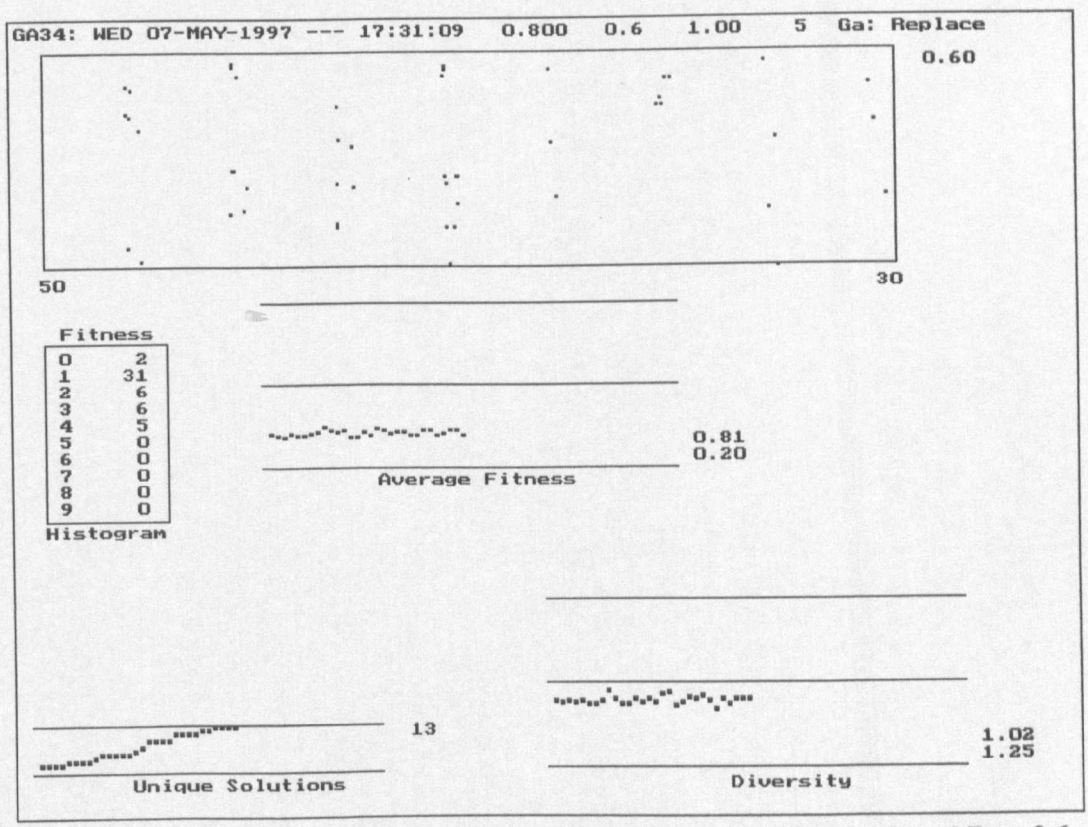

Figure 5-48: Test 18 GA Summary - Integer Chromosome with Pm = 0.8 and Pc = 0.6

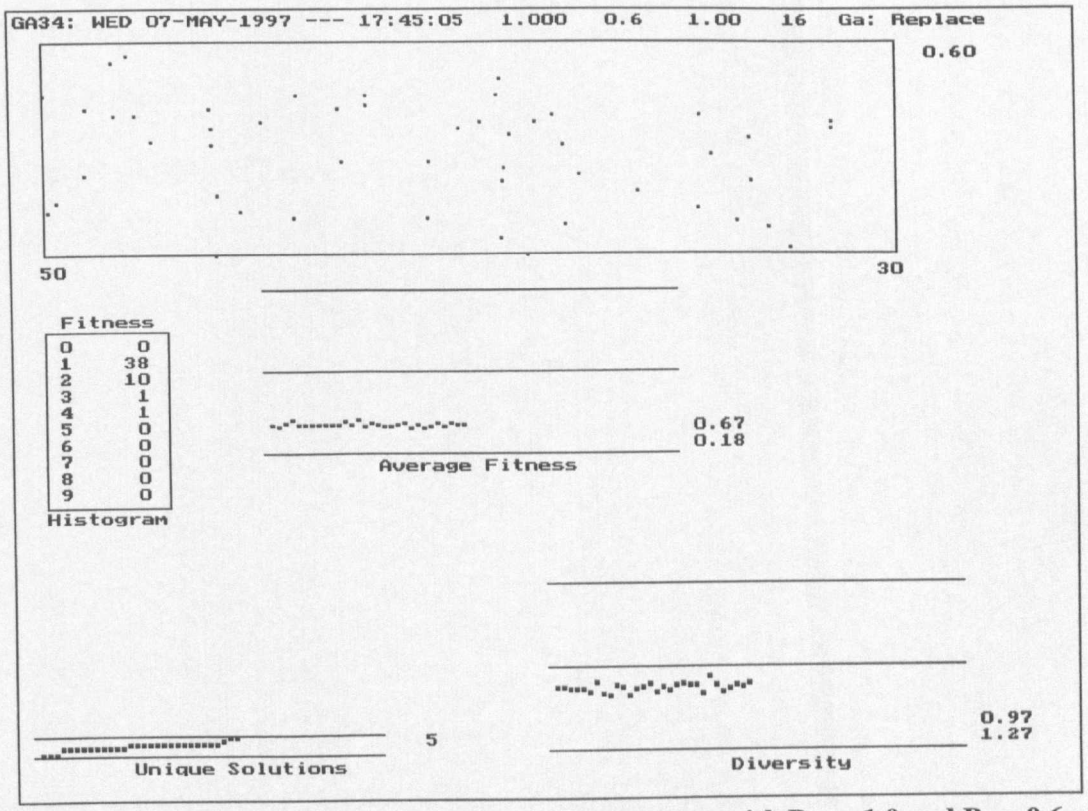

Figure 5-49: Test 19 GA Summary - Binary Chromosome with Pm = 1.0 and Pc = 0.6

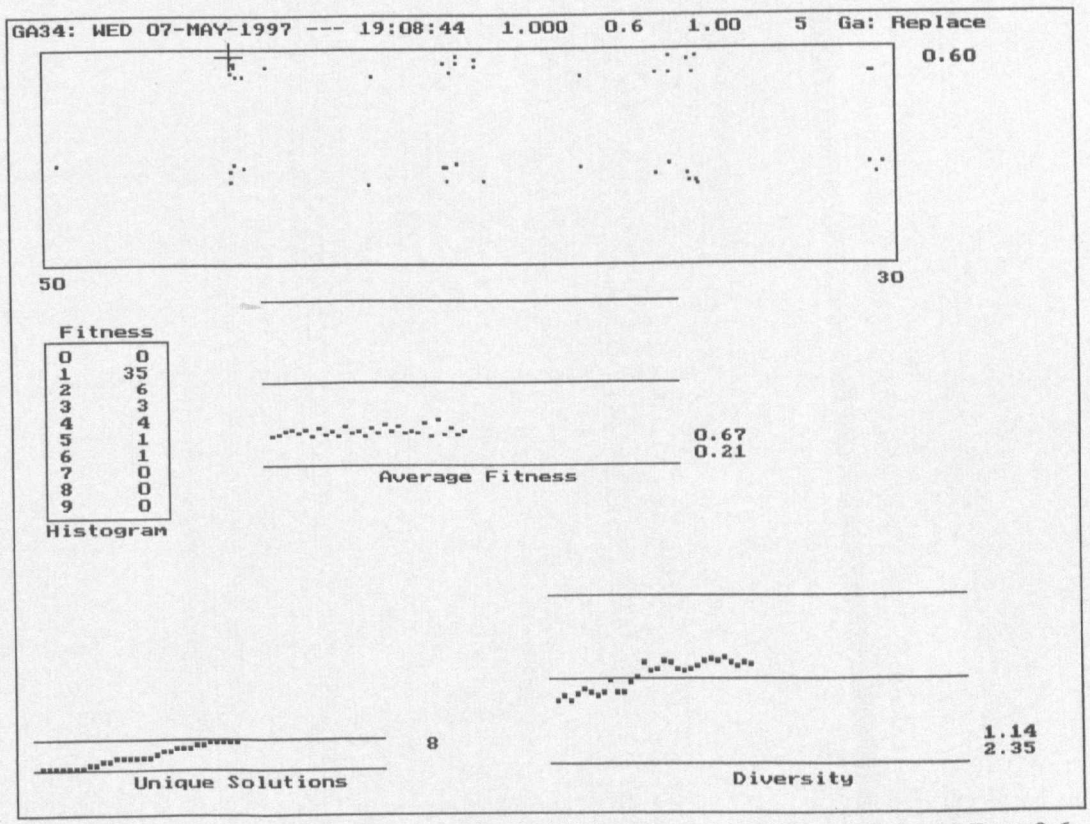

Figure 5-50: Test 20 GA Summary - Integer Chromosome with Pm = 1.0 and Pc = 0.6

The number of unique (individual) solutions, for a crossover probability of 0.6, decreased as the mutation probability increased (see Figure 5-30, Unique Solutions Column). For low values of mutation probability, a large number of similar ways of 'looking' at the contour were evolved, which had many different positions specified from where to start searching the list of segments from the PIP process. A variety of length thresholds were also evolved. For example, the fittest solution (Chromosome 1, Figure 5-10), for the digit two training set (Figure 5-8), had the observation genes in the chromosome equivalent to the combination 0-0-2, i.e. start at quadrant 0, look anti-clockwise and include 2 line-segments. The large number of solutions in this form all 'look' at the contour in the same way, but start 'looking' from different positions in the list of PIP line-segments and with a variety of line-segment length thresholds. Thus all the solutions were very similar and all 'look' at the contour in the same way. As the mutation probability was increased, the number of unique solutions decreased and these solutions were no longer clustered around the 'best' value. Also, as the mutation probability increased, the genetic algorithm search approached that of a purely random search.

From Figure 5-30 it can be seen that, for mutation probabilities greater than 0.2, the number of different ways of 'looking' at a contour remained approximately constant (Different Solutions Column). This suggests that the value of the mutation probability (greater than 0.2), for a given crossover probability, was not critical to performing the search for a variety of solutions. The number of 'good' solutions will be further discussed in Chapter 6 with regard to the suitability of these solutions to be used for contour recognition.

Figure 5-51: Test 21 and Figure 5-52: Test 22 show typical results for a genetic algorithm structured to randomly generate the chromosomes at the beginning of each generation, regardless of the crossover and mutation probabilities. The results displayed, in these two tests, suggest behaviour similar to a genetic algorithm with a mutation probability greater than 0.5. This type of result also indicates that, for mutation probability values greater than 0.5, the genetic algorithm was behaving

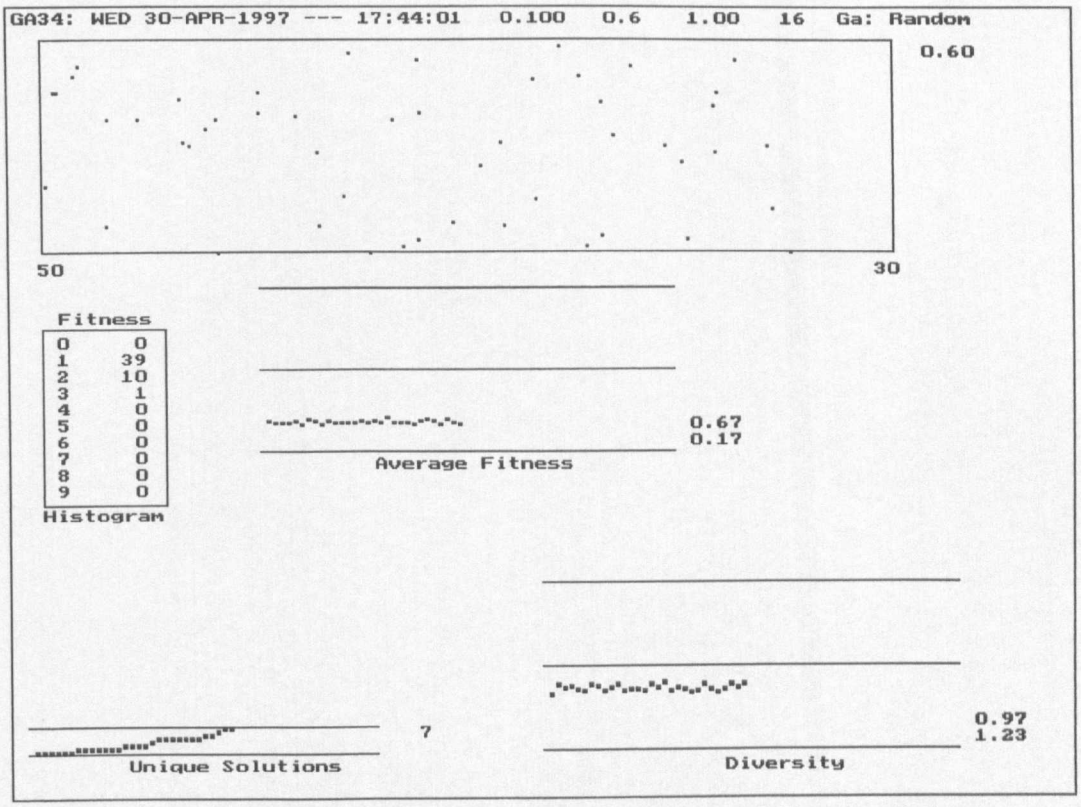

Figure 5-51: Test 21 GA Summary - Binary Chromosome with Random Selection

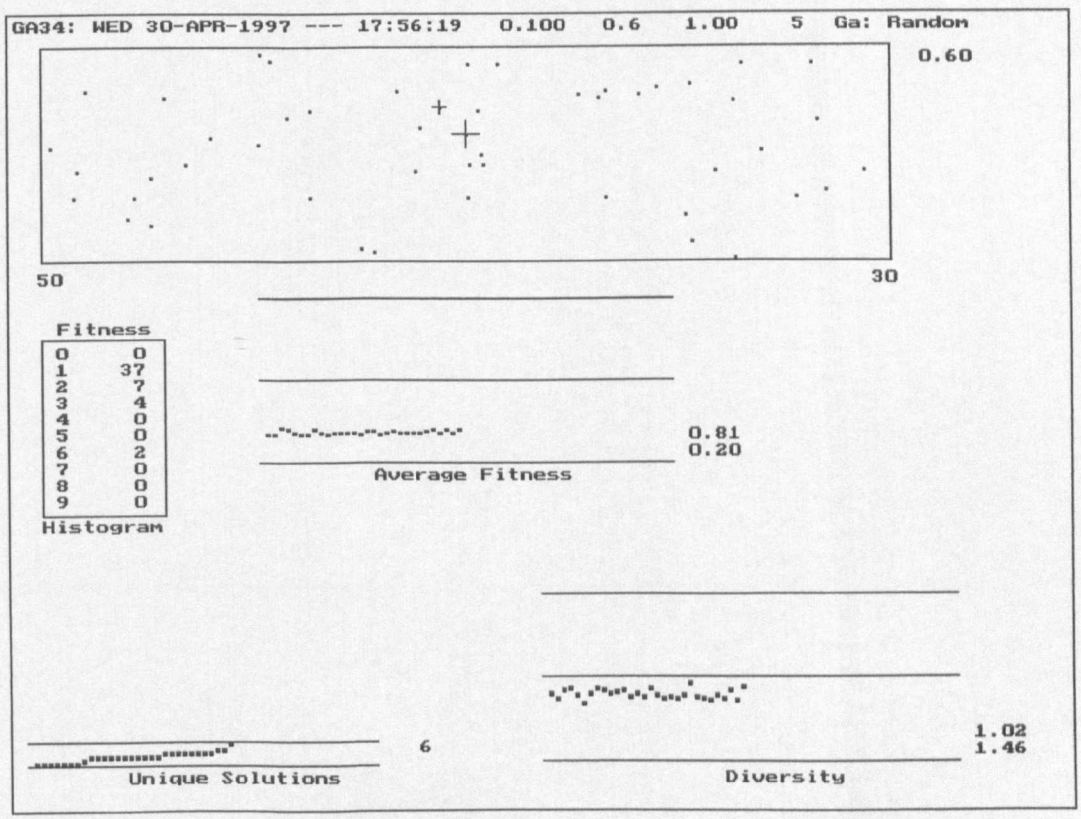

Figure 5-52: Test 22 GA Summary - Integer Chromosome with Random Selection

in a random fashion, with little contribution from the crossover operation.

Figure 5-53: Test 23 to Figure 5-62: Test 32 display the behaviour of the genetic algorithm with the crossover probability set to zero. The behaviour of the genetic algorithm was similar to that described above for mutation probabilities greater than 0.3. The diversity measure, in particular, followed the same pattern and reduces to a value below 2.0 for mutation probabilities greater than 0.25. These results suggest that mutation had a stronger influence on the behaviour of the genetic algorithm than crossover, when the crossover and mutation probabilities were set such that the genetic algorithm evolved a variety of solutions. The variation of chromosome diversity with mutation probability for these test cases is shown in Figure 5-26 and Figure 5-27.

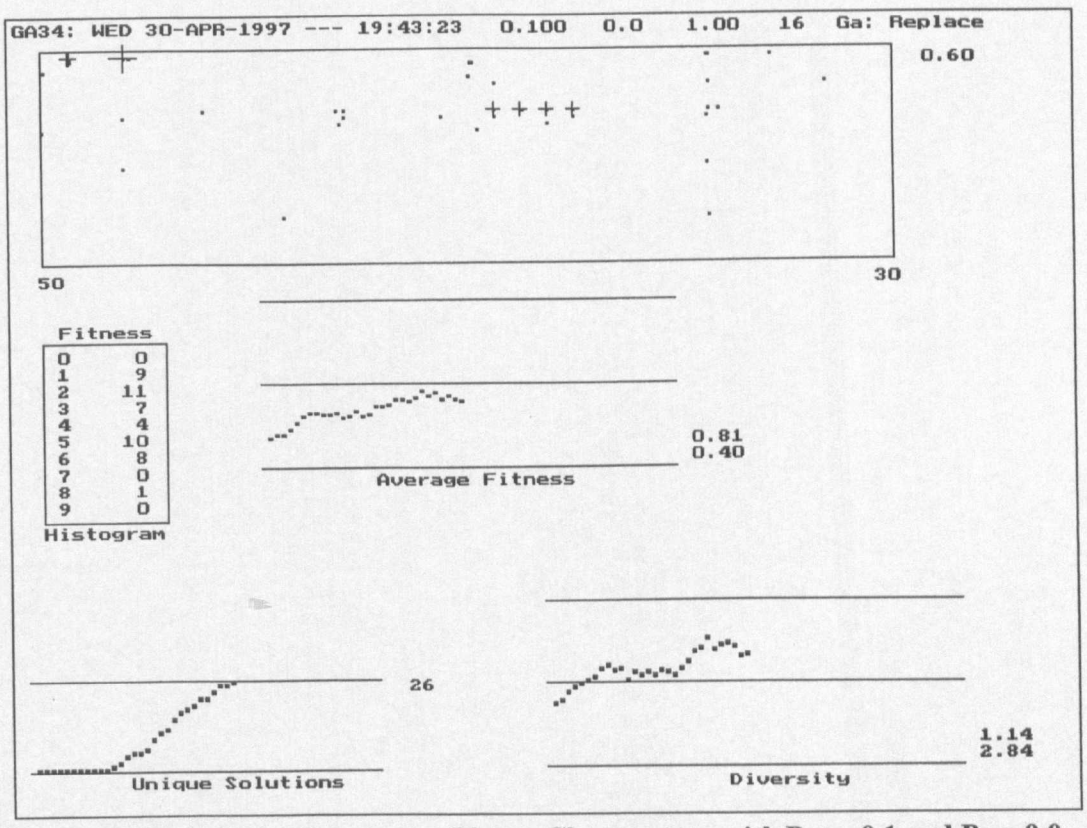

Figure 5-53: Test 23 GA Summary - Binary Chromosome with Pm = 0.1 and Pc = 0.0

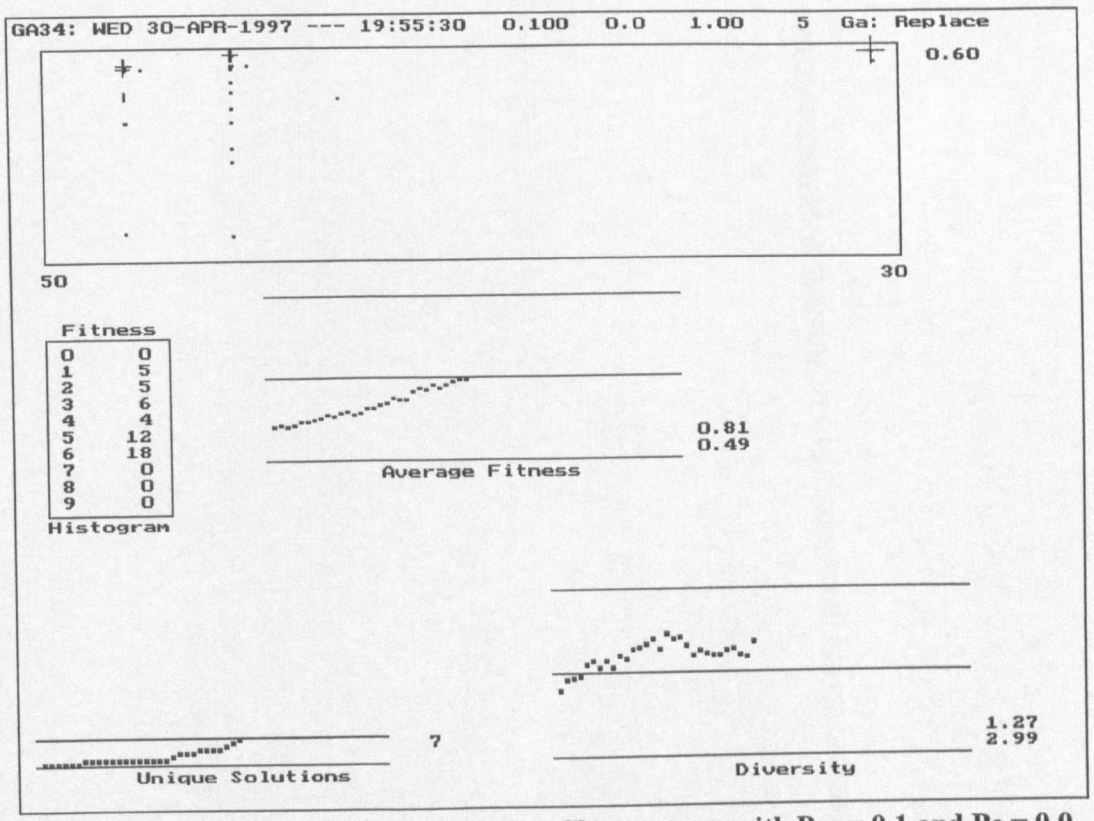

Figure 5-54: Test 24 GA Summary - Integer Chromosome with Pm = 0.1 and Pc = 0.0

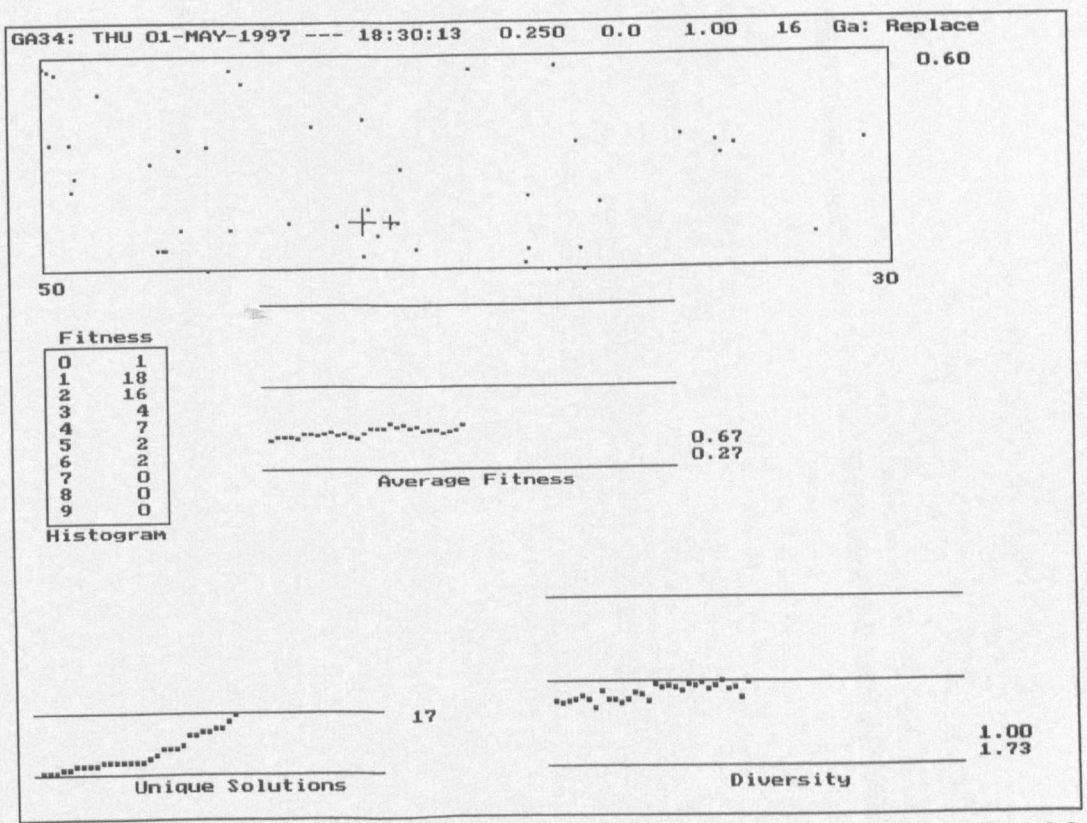

Figure 5-55: Test 25 GA Summary - Binary Chromosome with Pm = 0.25 and Pc = 0.0

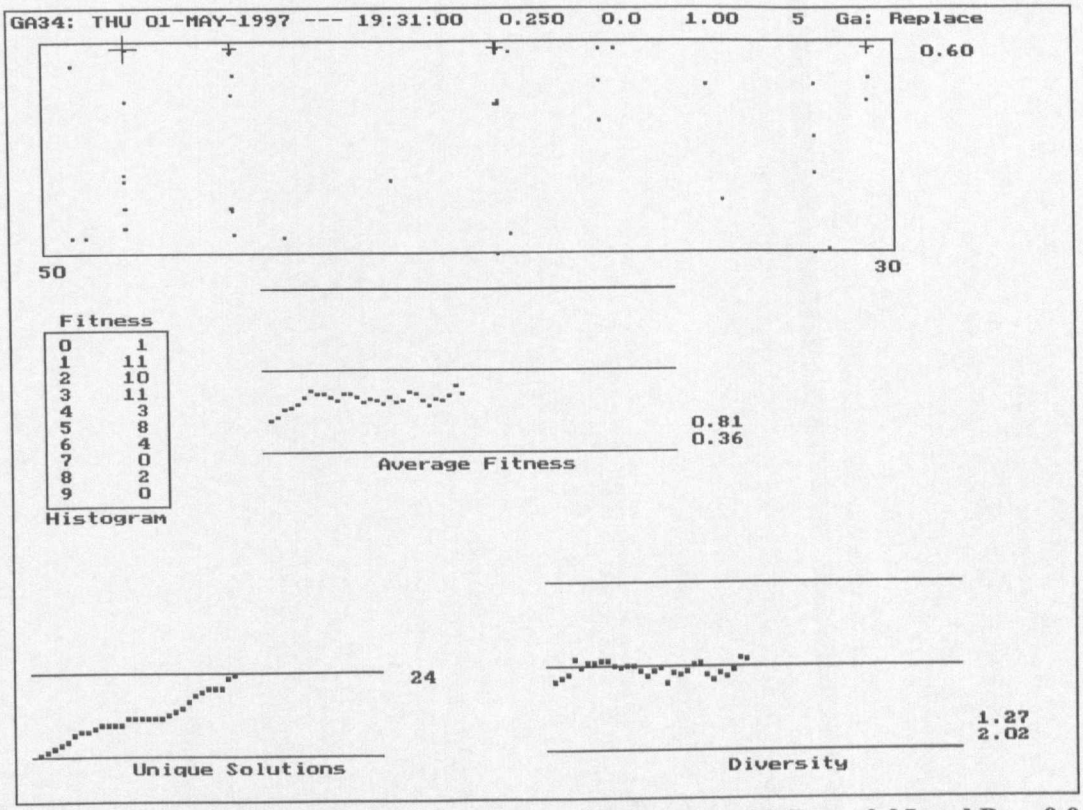

Figure 5-56: Test 26 GA Summary - Integer Chromosome with Pm = 0.25 and Pc = 0.0

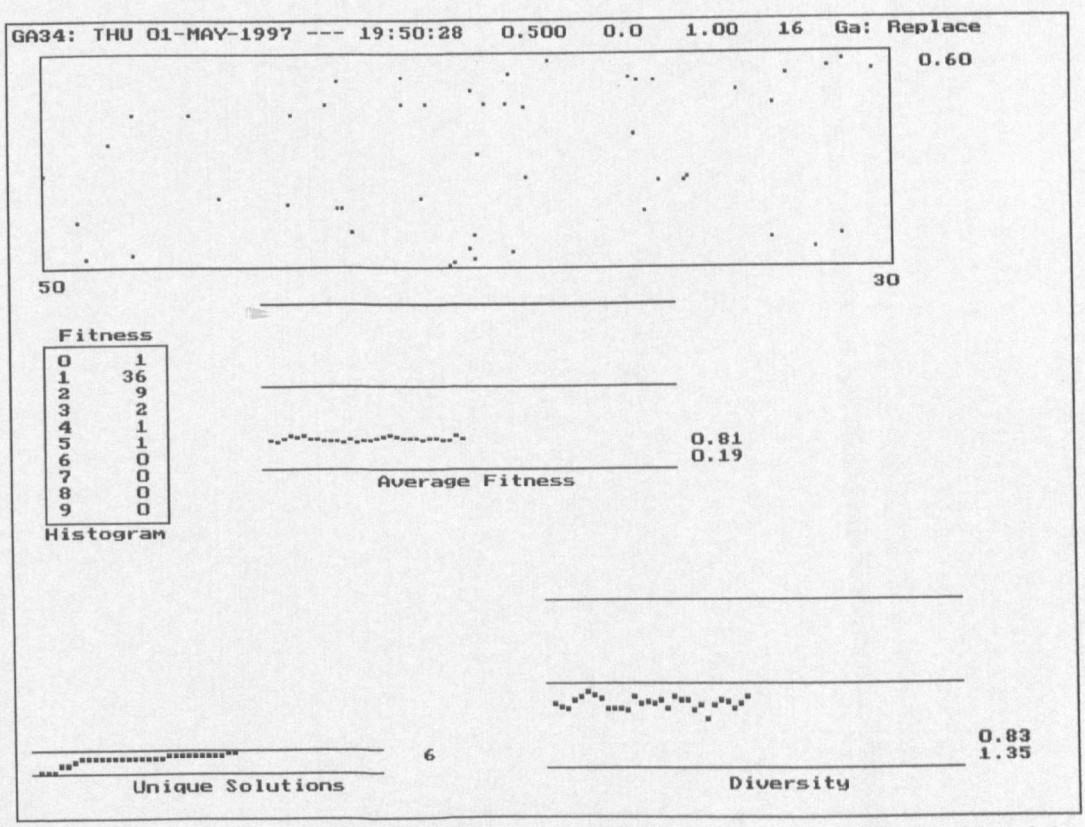

Figure 5-57: Test 27 GA Summary - Binary Chromosome with Pm = 0.5 and Pc = 0.0

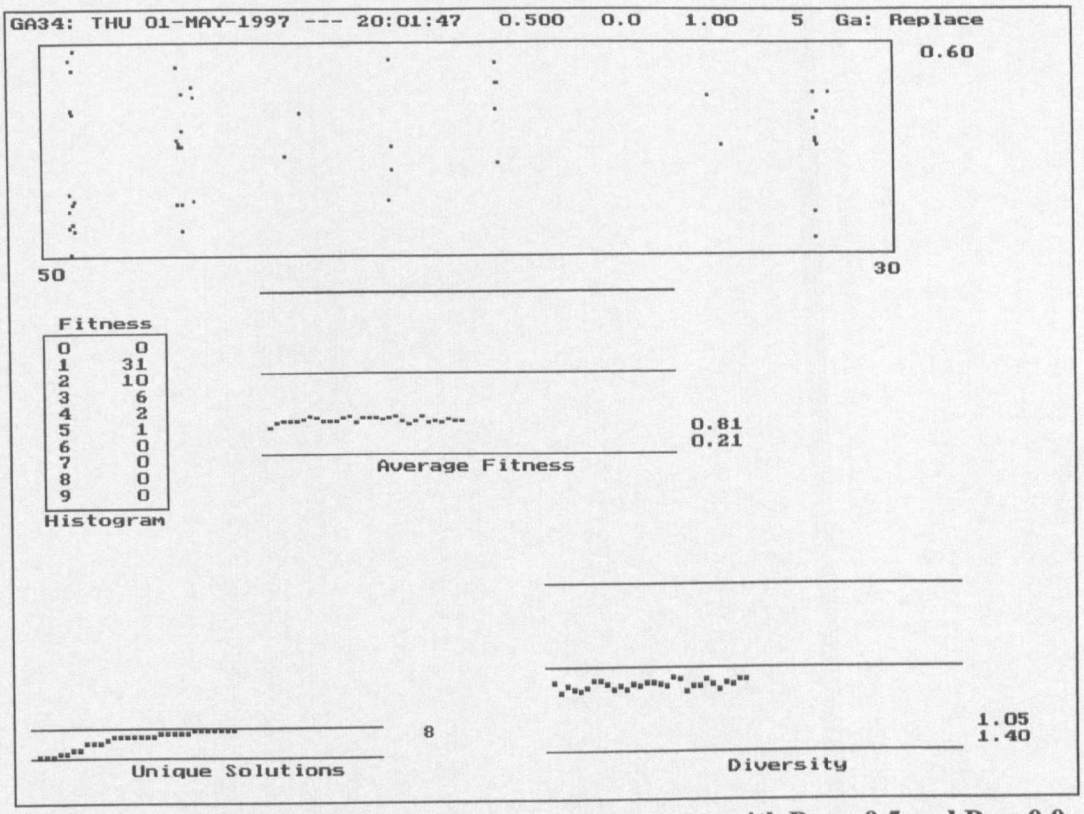

Figure 5-58: Test 28 GA Summary - Integer Chromosome with Pm = 0.5 and Pc = 0.0

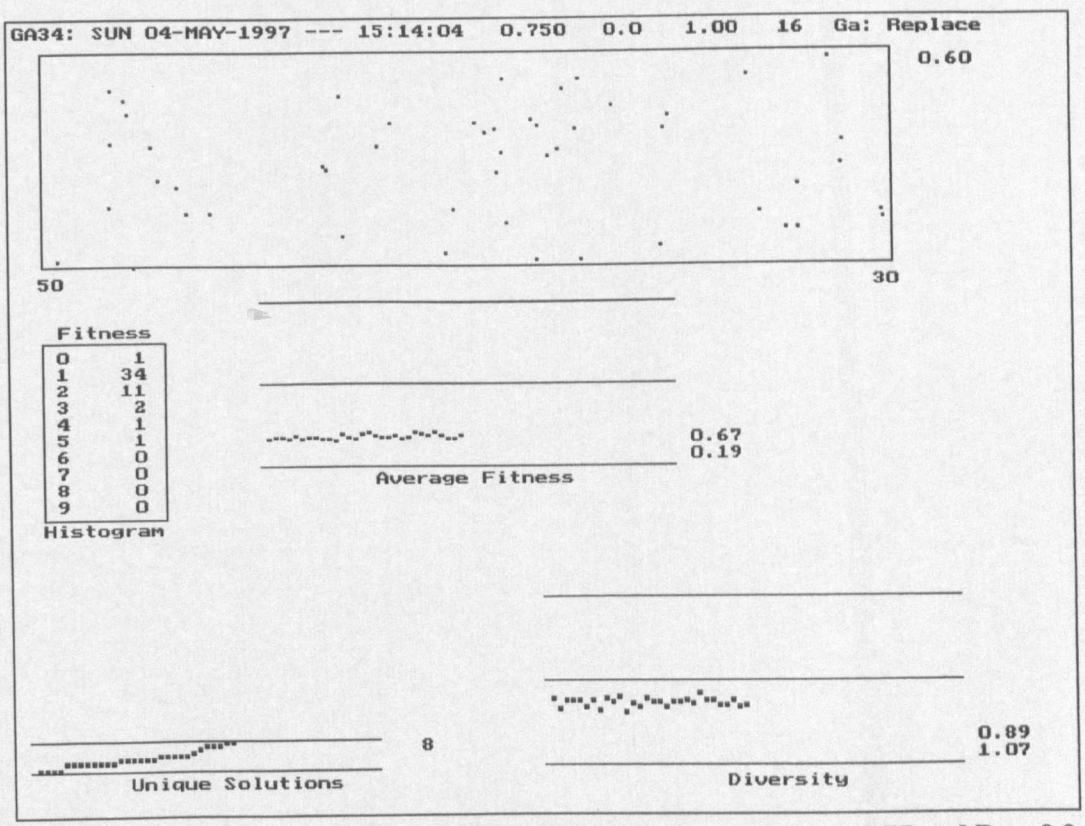

Figure 5-59: Test 29 GA Summary - Binary Chromosome with Pm = 0.75 and Pc = 0.0

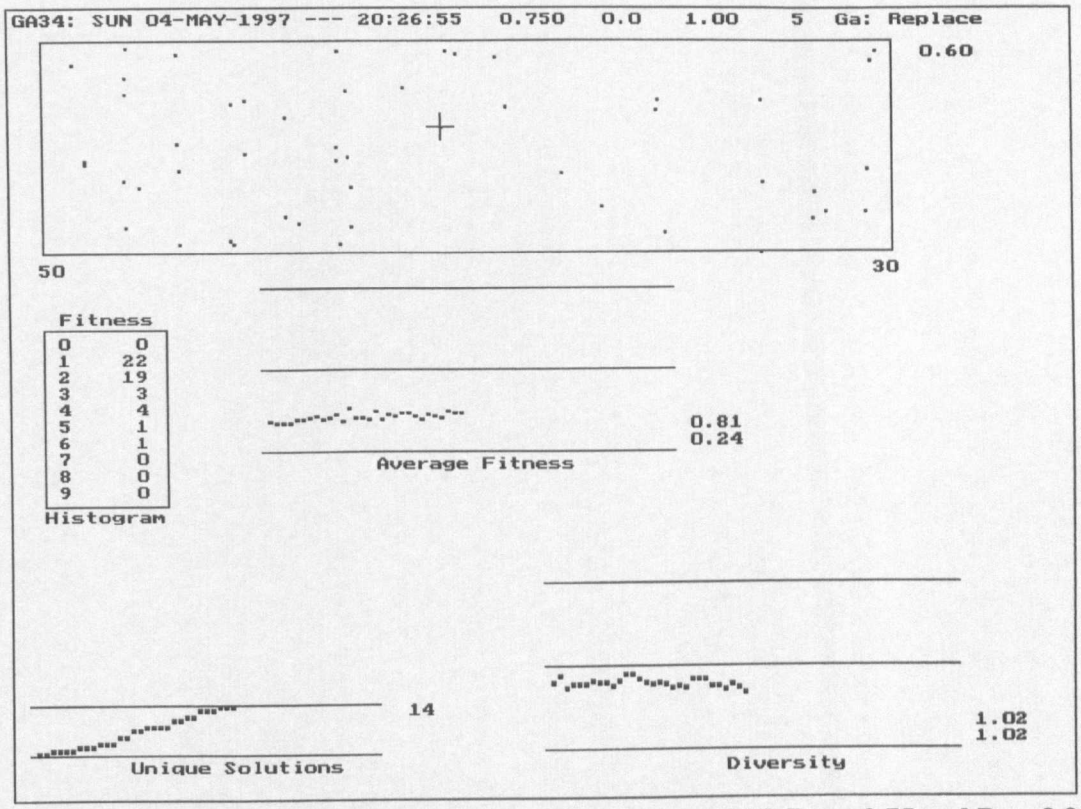

Figure 5-60: Test 30 GA Summary - Integer Chromosome with Pm = 0.75 and Pc = 0.0

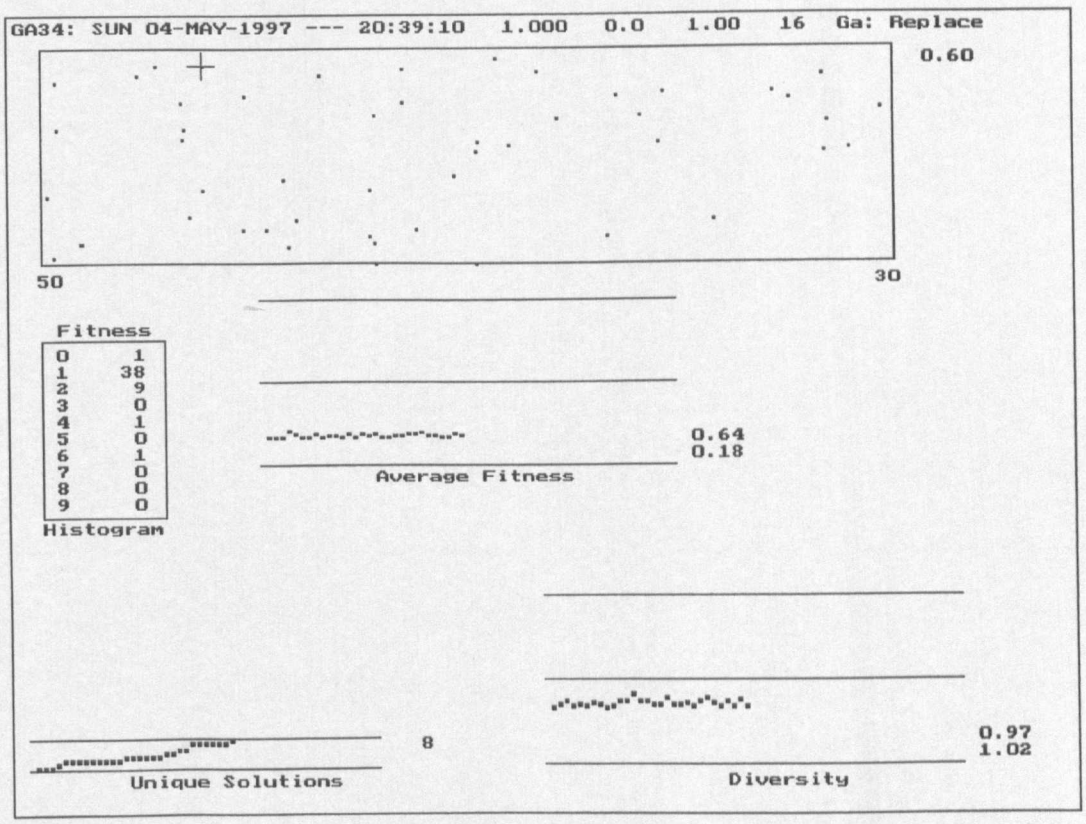

Figure 5-61: Test 31 GA Summary - Binary Chromosome with Pm = 1.0 and Pc = 0.0

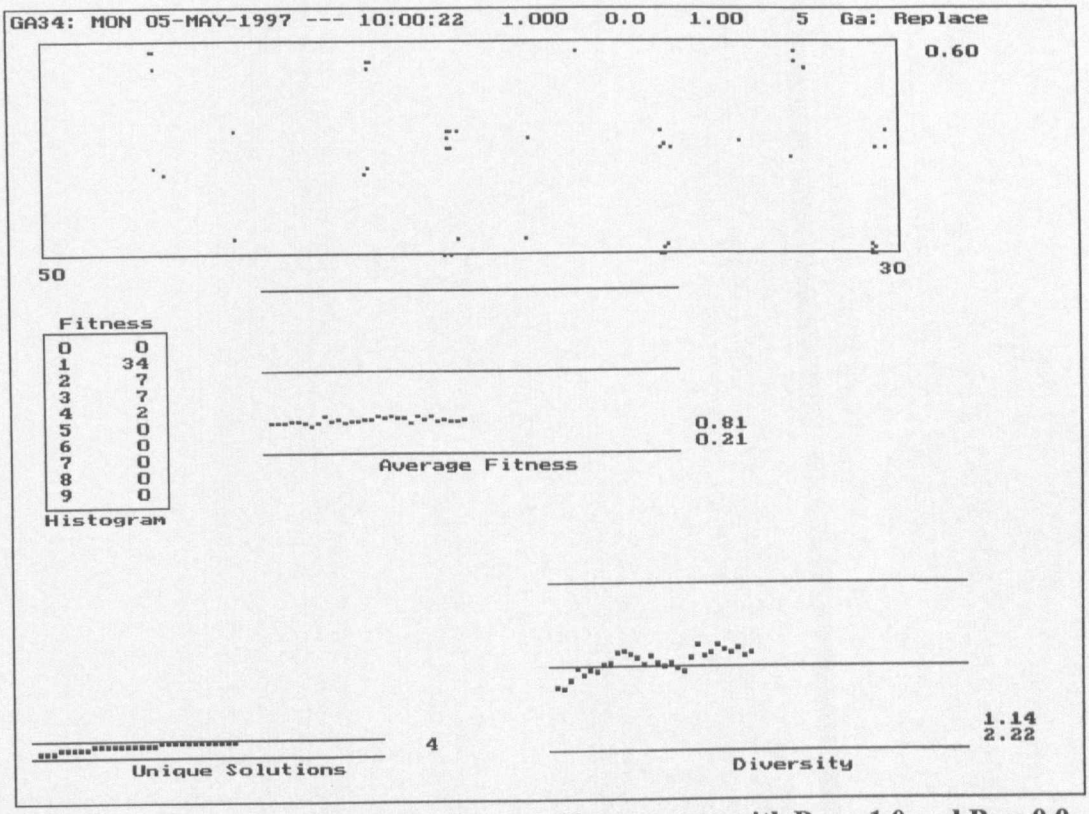

Figure 5-62: Test 32 GA Summary - Integer Chromosome with Pm = 1.0 and Pc = 0.0

Figure 5-63: Test 33 to Figure 5-72: Test 42 display the behaviour of the genetic algorithm with the mutation probability set to zero, i.e. only the influence of crossover by itself is observed. In general a small number of solutions were evolved which were not necessarily near to the 'best' solution of 0.81. These displays demonstrate a particular problem associated with the crossover operator. When the influence of crossover was too strong, the chromosomes evolved and became similar. Eventually the whole population became the same. At this stage in the evolution of a solution the crossover operator had no effect on chromosomes that were the same. Hence the evolution process stopped. In all cases the diversity measure indicated low diversity in the chromosome population and the average fitness increased to that of the 'best' evolved solution. The variation of chromosome diversity with crossover probability for these test cases is shown inFigure 5-28 and Figure 5-29.

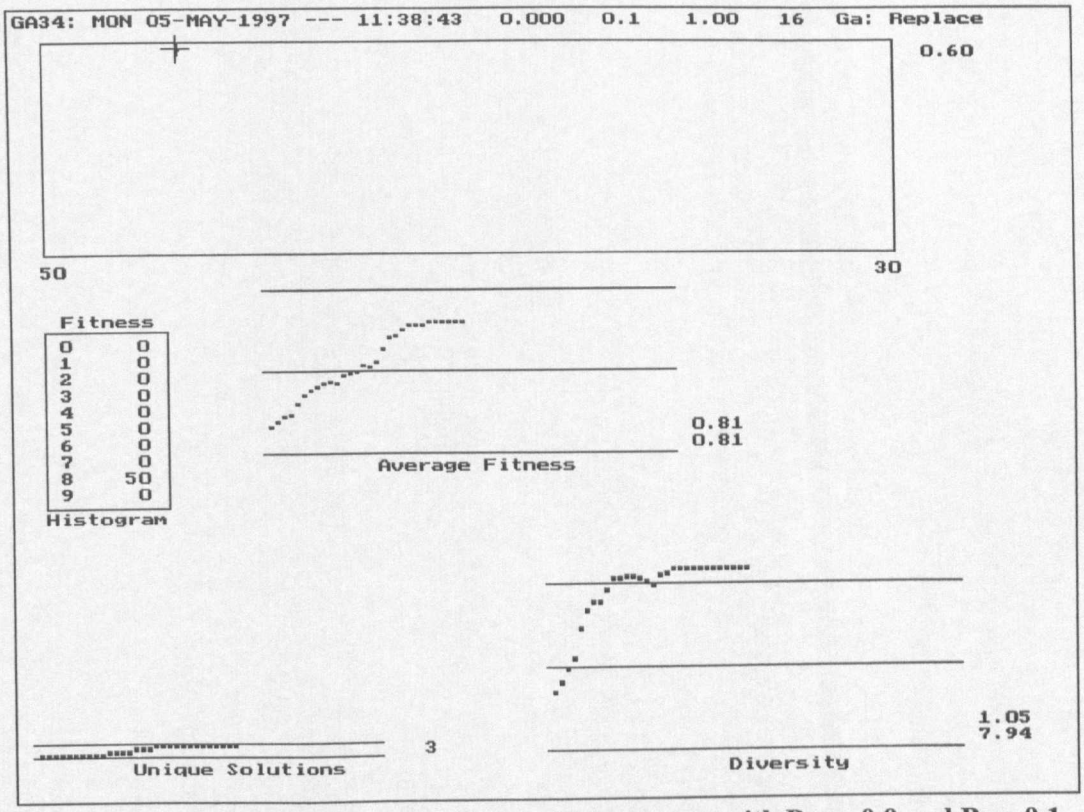

Figure 5-63: Test 33 GA Summary - Binary Chromosome with Pm = 0.0 and Pc = 0.1

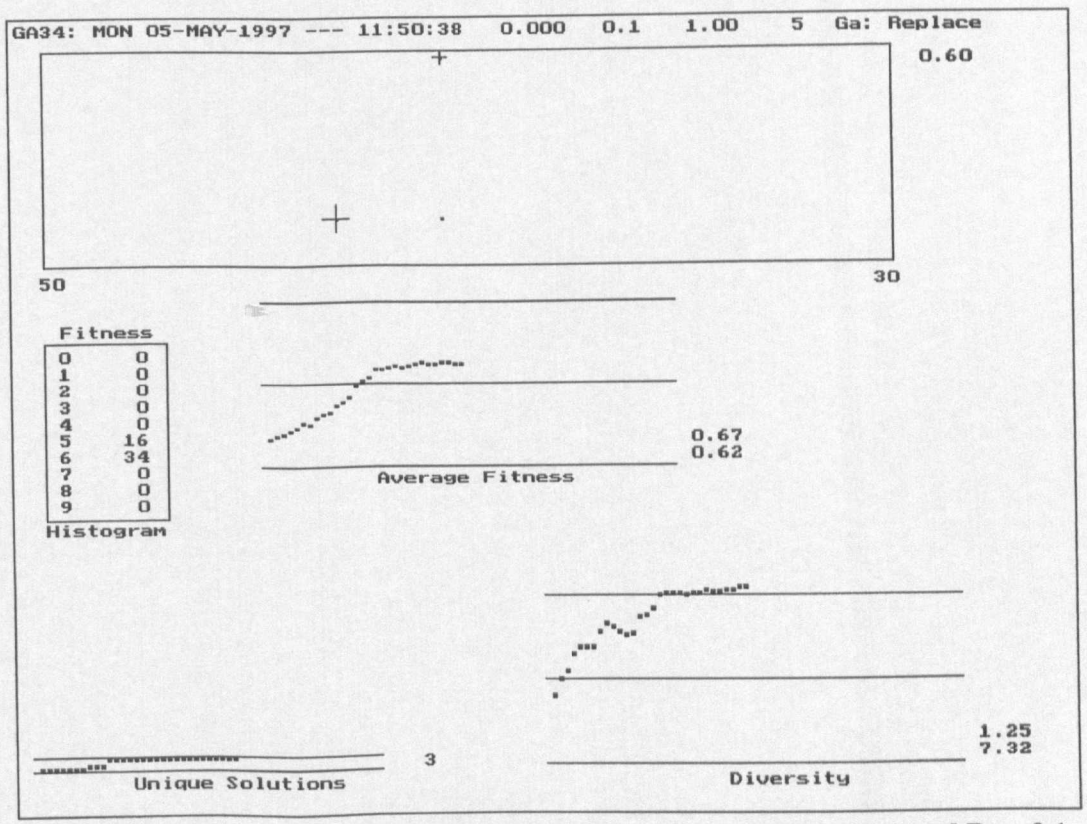

Figure 5-64: Test 34 GA Summary - Integer Chromosome with Pm = 0.0 and Pc = 0.1

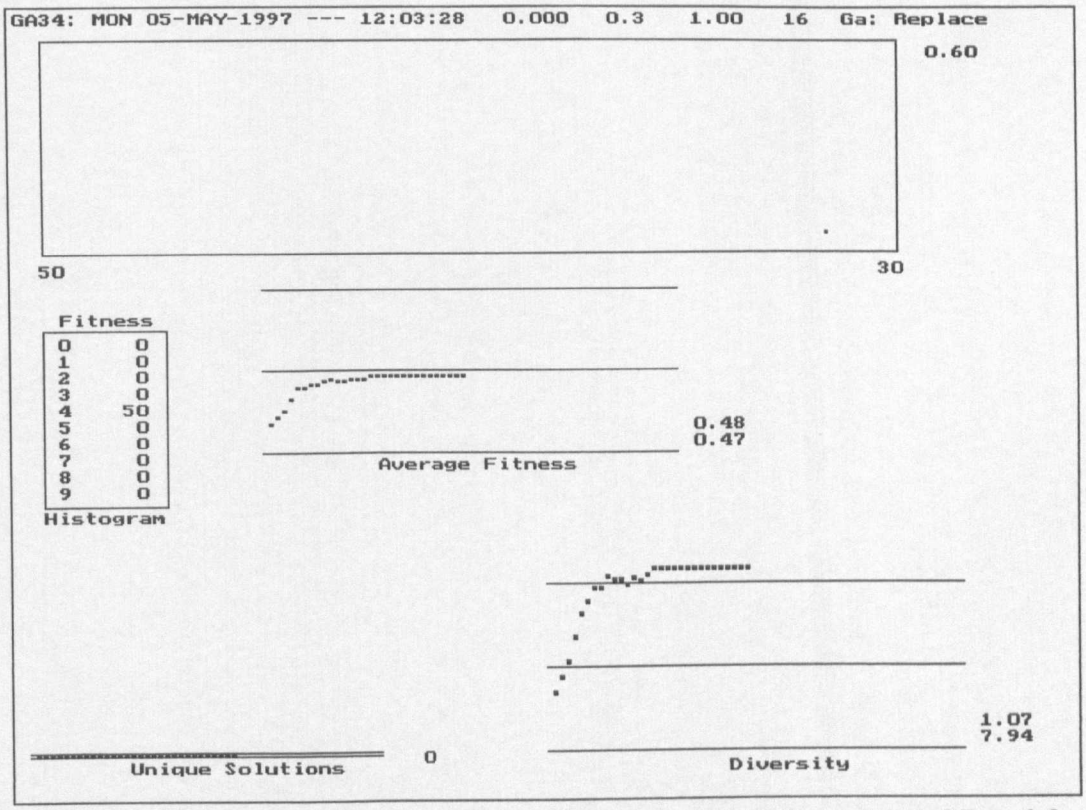

Figure 5-65: Test 35 GA Summary - Binary Chromosome with Pm = 0.0 and Pc = 0.3

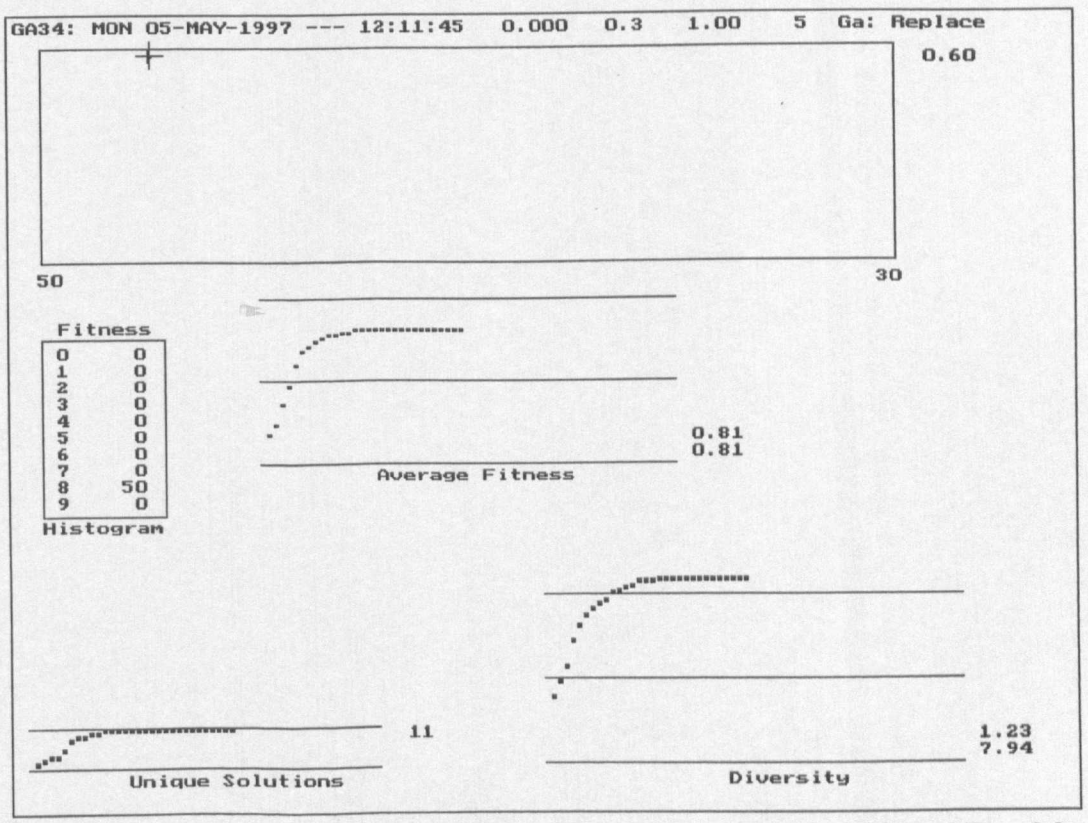

Figure 5-66: Test 36 GA Summary - Integer Chromosome with Pm = 0.0 and Pc = 0.3

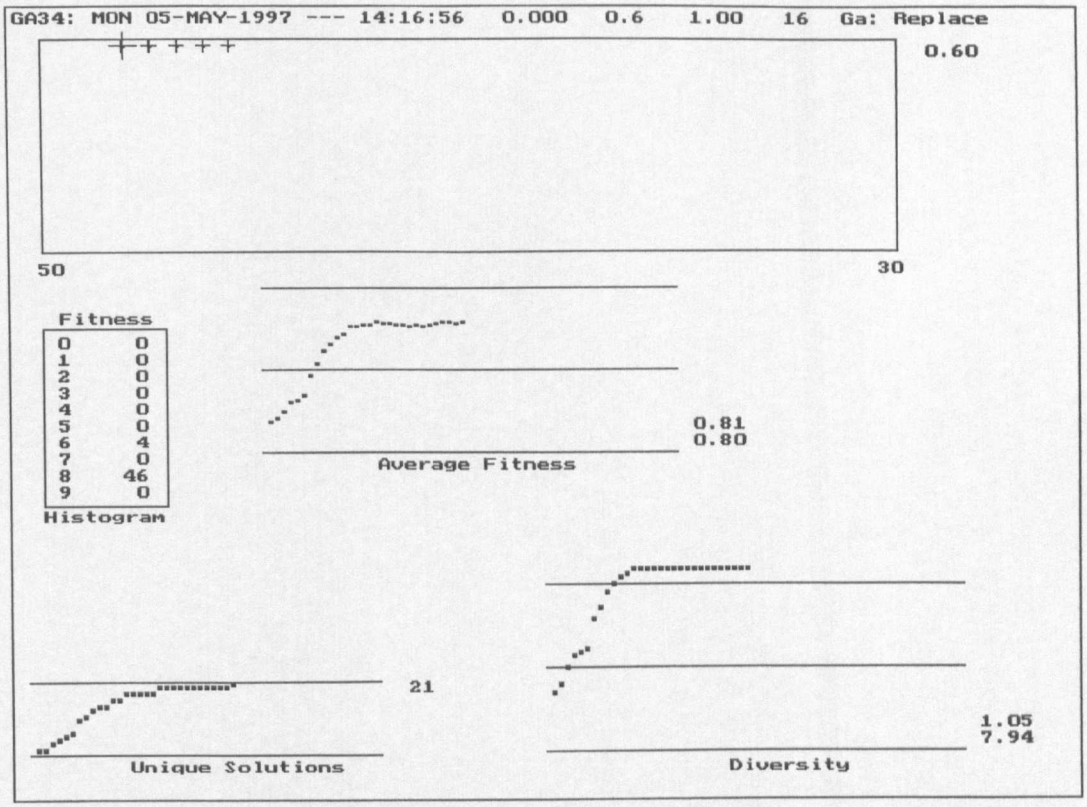

Figure 5-67: Test 37 GA Summary - Binary Chromosome with Pm = 0.0 and Pc = 0.6

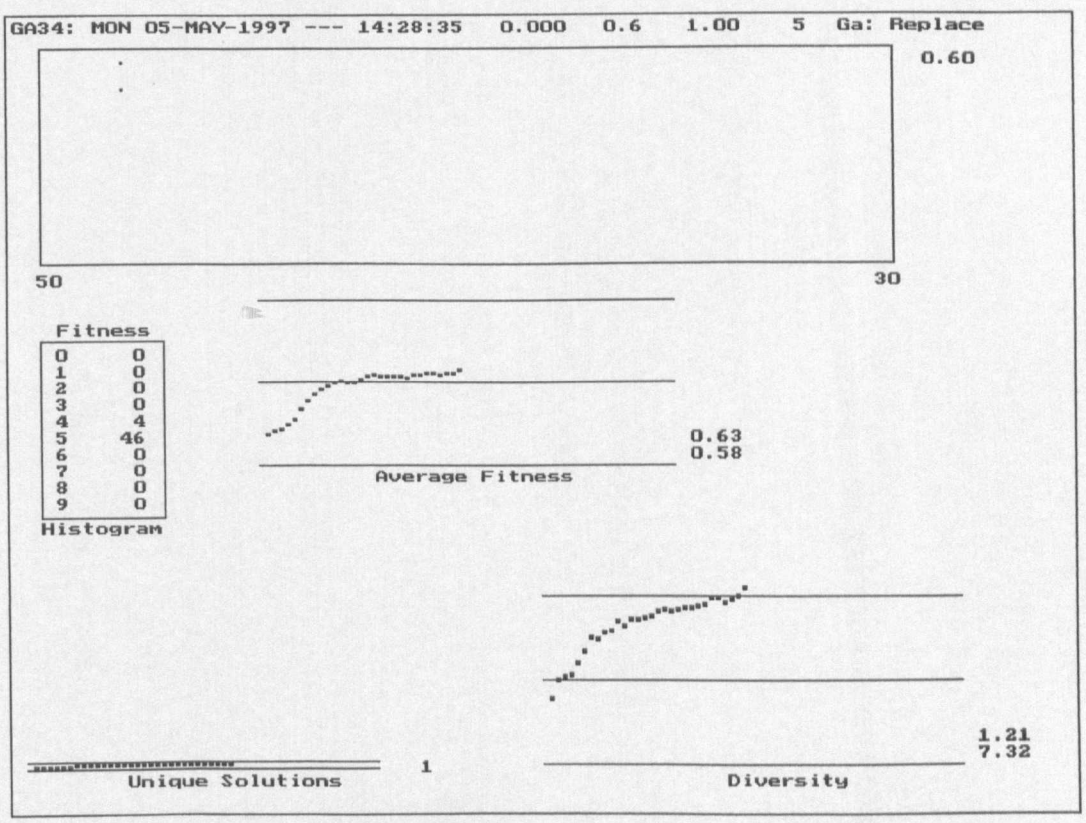

Figure 5-68: Test 38 GA Summary - Integer Chromosome with Pm = 0.0 and Pc = 0.6

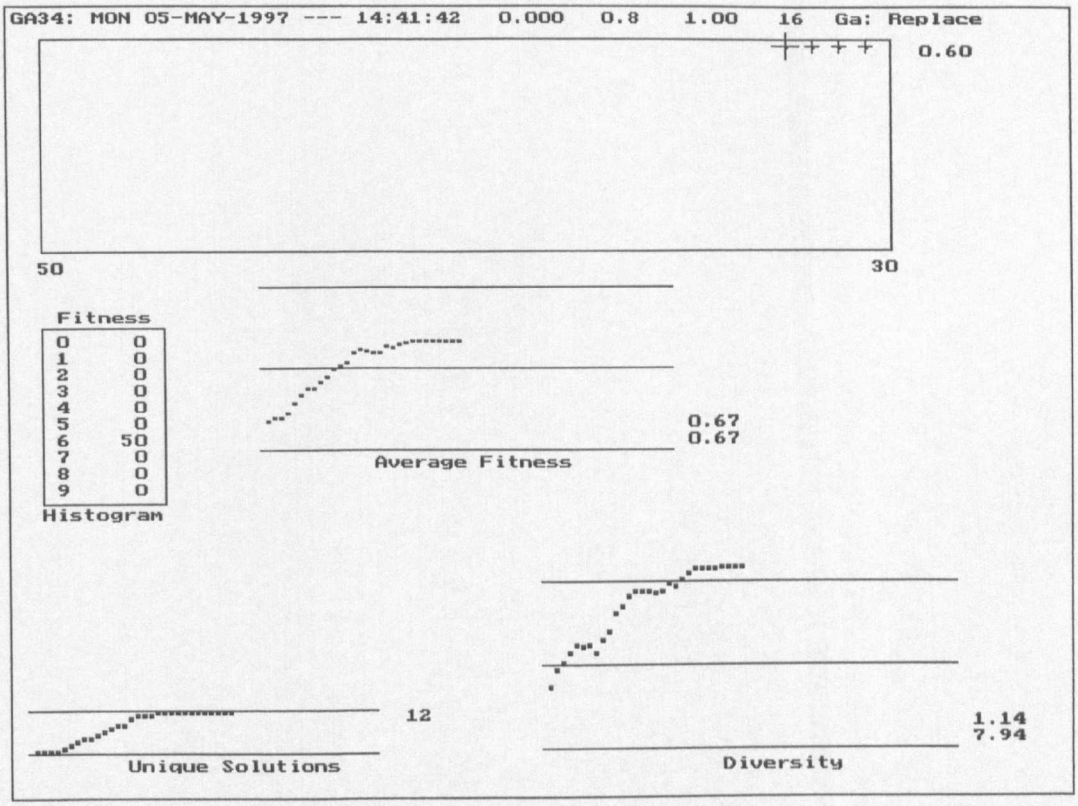

Figure 5-69: Test 39 GA Summary - Binary Chromosome with Pm = 0.0 and Pc = 0.8

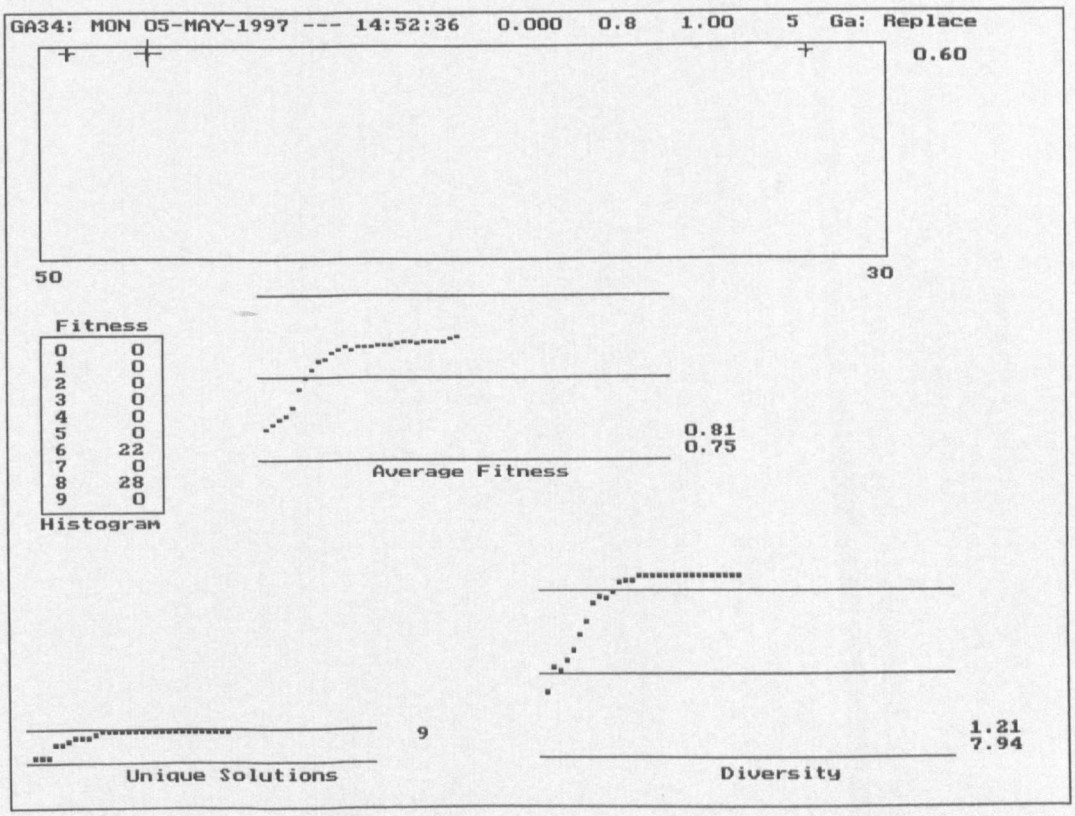

Figure 5-70: Test 40 GA Summary - Integer Chromosome with Pm = 0.0 and Pc = 0.8

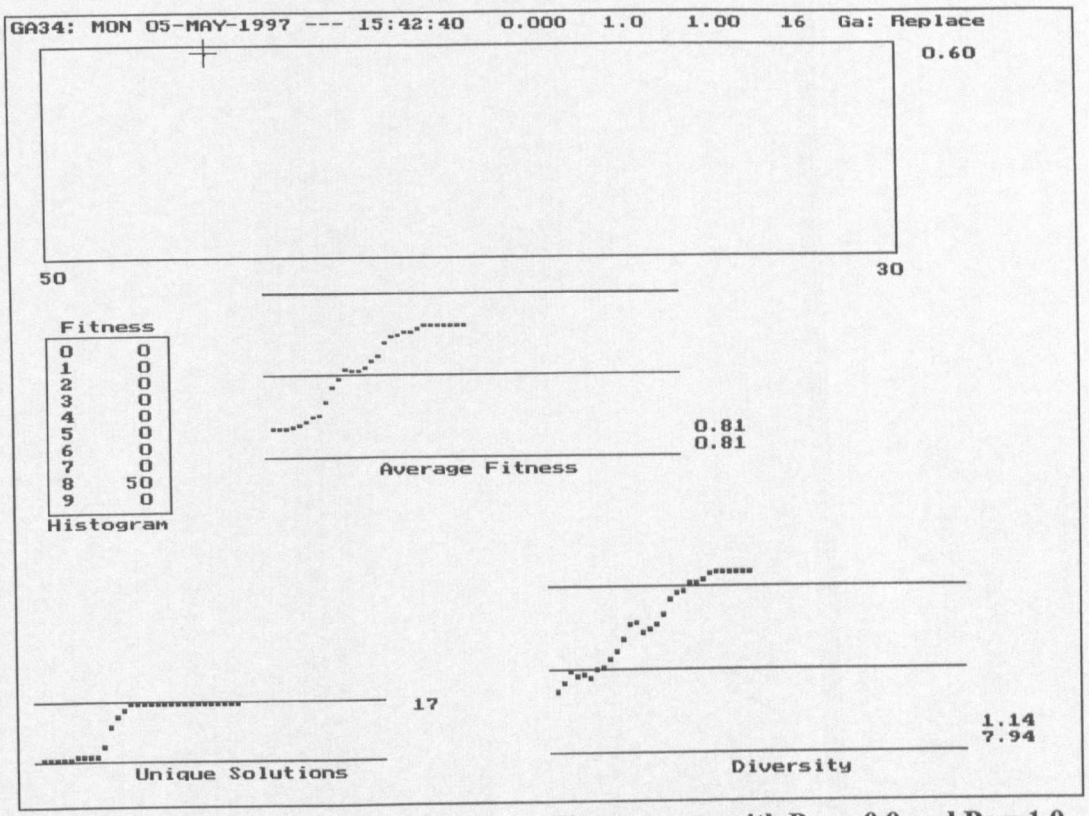

Figure 5-71: Test 41 GA Summary - Binary Chromosome with Pm = 0.0 and Pc = 1.0

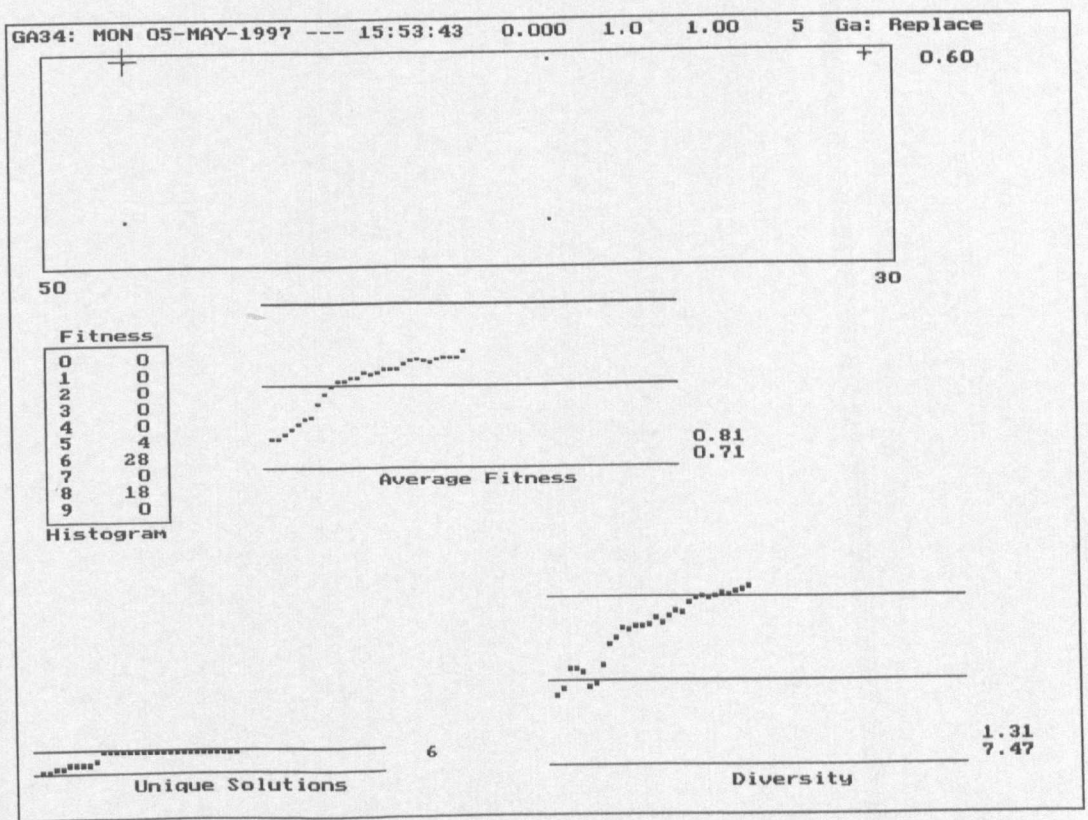

Figure 5-72: Test 42 GA Summary - Integer Chromosome with Pm = 0.0 and Pc = 1.0

Figure 5-73: Test 43 and Figure 5-74: Test 44 show typical results for the behaviour of the genetic algorithm for the condition where the search was for a 'mismatch' rather than a match. The chromosomes evolved in this mode specify how to observe a contour so that the combinations of PIP line-segment vectors indicated how the examples in the digit two training set differed from each other. This information complemented the matching information evolved in the normal mode of operation for the genetic algorithm. Note that a 'mismatch' was obtained when the chromosome had low fitness. When the fitness of the chromosome, i.e. the fitness used by the genetic algorithm for selection of a chromosome for crossover and mutation, was set to (1.0 - the calculated fitness), the genetic algorithm evolved chromosomes that have low measured fitness. The fitness threshold was increased to 0.9 so that the amount of chromosome information stored onto disc for use by the recognition process was reduced (see also Chapter 6).

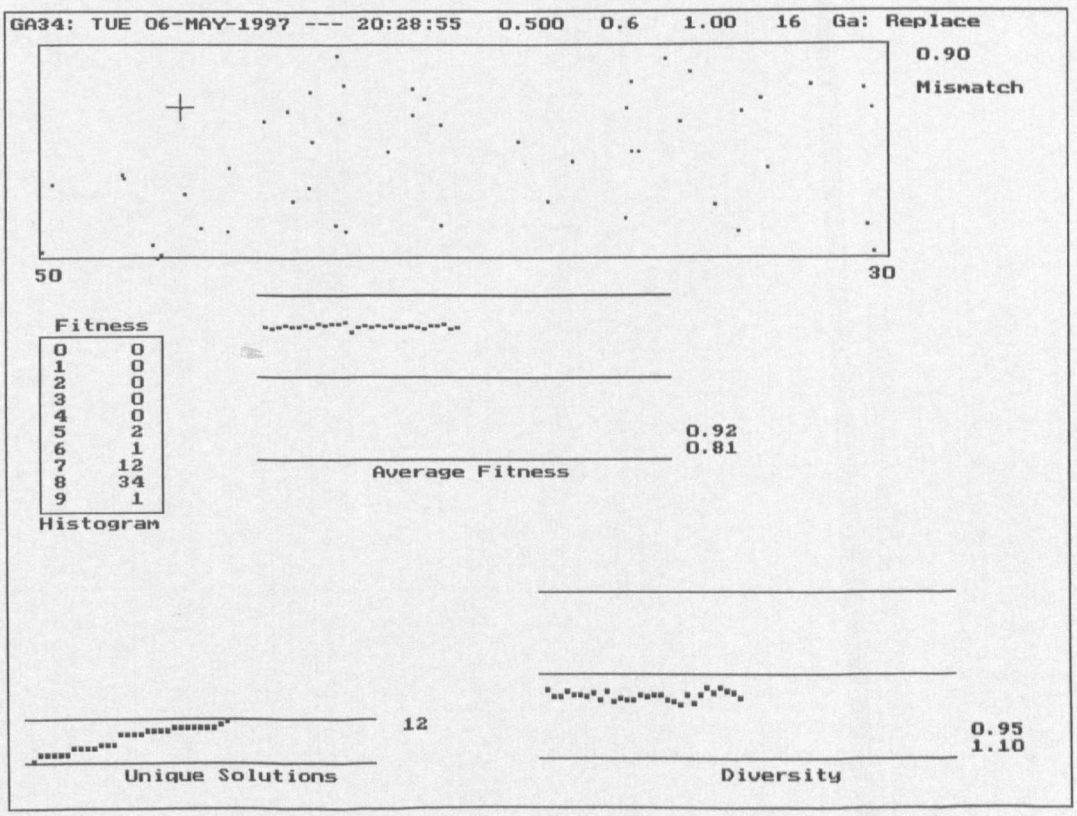

Figure 5-73: Test 43 GA Summary - Binary Chromosome with Pm = 0.5 and Pc = 0.6

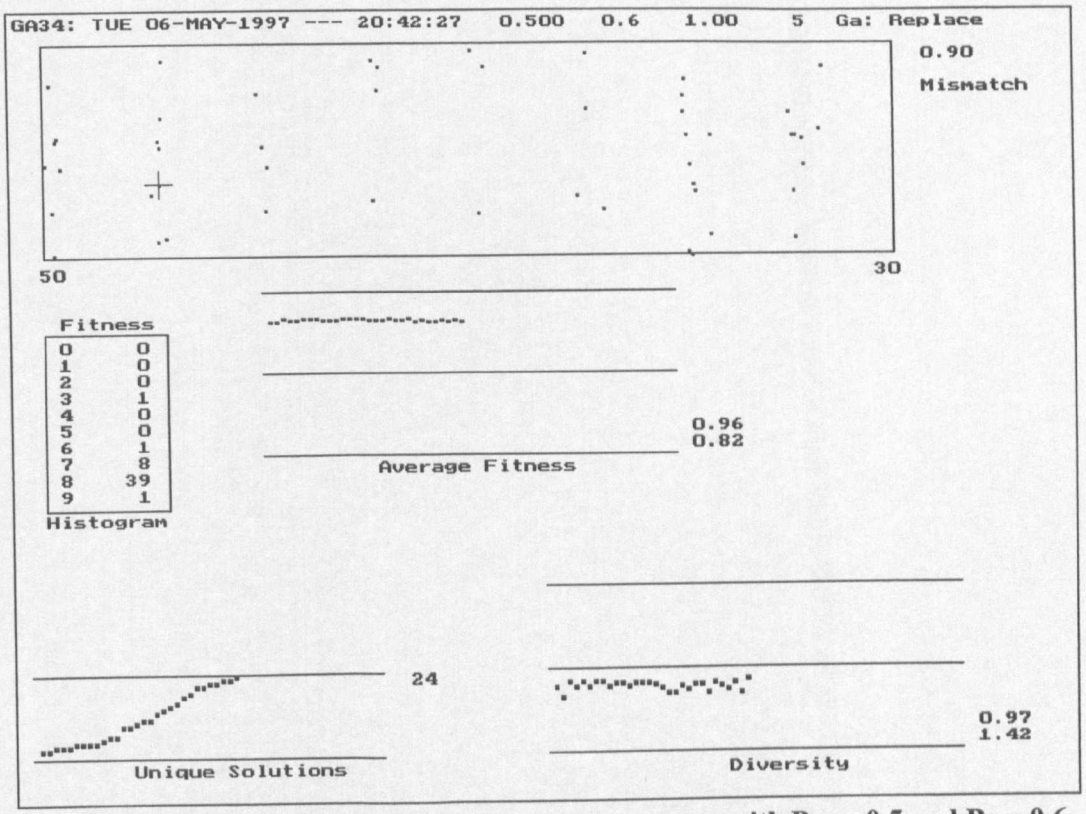

Figure 5-74: Test 44 GA Summary - Integer Chromosome with Pm = 0.5 and Pc = 0.6

5.9 Summary and Conclusions

The research discussed in this chapter has designed and investigated a version of the standard genetic algorithm that used binary and integer chromosomes. Chromosomes were chosen for crossover and mutation in pairs by 'roulette wheel' selection, where the probability of choice is proportional to the fitness of the chromosome. The chromosome was structured such that the various genes specify how the PIP line-segment vectors are to be observed or 'looked at' as groups. A variety of ways to observe a contour with high fitness were evolved using a training set. The fittest ways to 'look at' a contour should be able to be used to recognise a test contour. The research method used for the recognition process is described in Chapter 6. A sixteen-bit binary and a five-integer chromosome were implemented, thus allowing for 65536 solutions. Typically a population of fifty chromosomes evolving for thirty generations was used for most of the experimental tests.

A triangular fitness function was used for these tests and the triangular function was centred on the

mean value for a PIP line-segment parameter (e.g. length of the training set PIP line-segment vectors specified by the relevant gene in the chromosome). The slope of the triangular fitness function was adjusted by the value of the standard deviation of the appropriate PIP line-segment parameter. The training set was comprised of five examples of the digit two drawn by the author and five examples from a database of numerals used by one of the Open University Artificial Intelligence Technology Courses (T396 Block 3 Technology Third Level).

The genetic algorithm could also be run using a 'mismatch' fitness function, i.e. (1.0 – calculated fitness) so that groups of PIP line-segment vectors were evolved that indicated which parts of the training set contours did not match.

With the crossover probability set to the value 0.6 and the mutation probability set to values less than 0.2, the fittest solution (fitness = 0.81) was developed with the average fitness increasing as the evolution proceeded. In general, a selection of 'good' solutions was required for the contour recognition process, so that the variation contained in the training set contours could be adequately covered by the different ways to 'look at' each contour as specified in each chromosome. The fittest solutions for these values of the crossover and mutation probabilities specified, the same start quadrant, the same direction to search the list of PIP line-segment vectors and the same number of PIP line-segments, e.g. start quadrant 0, search anti-clockwise and calculate the fitness for 2 PIP line-segment features. The main variations in the chromosomes were the genes that defined where to start looking in the PIP line-segment list and the line-segment length threshold. These tests showed that the genetic algorithm performed in a similar manner to a standard genetic algorithm and exhibited the usual characteristics for the standard genetic algorithm reported in the literature. The genetic algorithm appeared to have the same performance when using binary or integer chromosomes.

This research work has shown that in order to search the solution space for a selection of 'good'

solutions the mutation probability had to be increased to values in the region of 0.2. The genetic algorithm search behaved more randomly as the mutation probability was increased towards the value 1.0. Under these circumstances the genetic algorithm appeared to perform better than a random search, due to the action of the crossover operator.

A new chromosome diversity measure has been developed during the research work described in this chapter. This new diversity measure uses the standard deviation of the observation genes in the chromosome to calculate the spread in the various ways of 'looking at' a contour specified by these observation genes. The performance of this diversity measure has been compared to one from the literature (Lis, 1995). The results obtained from these two methods appeared to be consistent. The new diversity method may have advantages because it appeared to have a sharper 'cut-off', at mutation probability values around 0.2, than the fitness dispersion method (see also Lis, 1995).

The research has shown that the three measures of *diversity*, *number of unique (individual)* solutions and *number of different* solutions can be used to choose the value of the mutation probability such that a variety of different solutions are evolved. The preferred value for the crossover probability was chosen as 0.6 as suggested in the genetic algorithm literature.

Further processing of the solutions such as grouping of the solutions by start quadrant could possibly be used to provide a certain amount of immunity to partial occlusion of a contour. This grouping may also help when different contour shapes have sections that are similar. The genetic algorithm was normally executed for 30 generations with a population of 50 chromosomes. Therefore 1500 fitness calculations were typically performed to obtain a selection of solutions (total number of solutions equals 65536). Hence the search was efficient and could possibly be used with a network of processors to perform the training exercise on a sequence of real-time images.

The research has also shown that the standard genetic algorithm parameters can be adjusted to

evolve either

1. A single 'best' way to 'look at' a training set of contours that may be unique to that particular training set or
2. A selection of 'good' ways to observe the training set of contours that can describe the contours more thoroughly.

Measurement of the diversity of the individual genes in a chromosomes and/or the monitoring of the fitness dispersion of the chromosome population has been shown to have potential use in choosing the values of the crossover and mutation probabilities to evolve a selection of 'good' solutions.

Krishnakumar (1989) has pointed out that one drawback of the standard or canonical genetic algorithm is the time penalty involved in evaluating the fitness functions for large populations, and explored a small population approach (given the name micro-genetic algorithm) with some very simple genetic parameters. It was shown that the micro-genetic algorithm reached the near-optimal region much earlier than the standard genetic algorithm. The population size was set to five and the fittest chromosome was passed to the next generation automatically (an elitist strategy). A tournament selection strategy was used by Goldberg and Deb (1991) to randomly group the five chromosomes and then adjacent pairs were chosen for reproduction. A crossover operator was applied with a probability of 1.0 and the mutation probability equals 0.0. The evolutionary process was stopped on a 'reasonable' measure of convergence based on either genotype or phenotype behaviour. The micro-genetic algorithm described in Krishnakumar (1989) was applied to the real-world non-stationary problem, such as those related to the wind shear optimal guidance problem, that is of keen interest to aircraft manufacturers regarding aircraft safety. This type of micro-genetic algorithm format could be applied to the genetic algorithm discussed in this chapter and is suggested, in Chapter 9, as the subject of further research.

Chapter 6 investigates how to use the chromosomes evolved by the genetic algorithm developed in

this chapter and the triangular fitness function used by this genetic algorithm to recognise other similar shaped contours.

6. Contour Shape Recognition Using Chromosome Encoding

6.1 Introduction

The genetic algorithm used in the research discussed in Chapter 5 evolved a list of chromosomes that were able to specify how to observe various examples of a shape contour, such that certain groups of the line-segment vectors around the contour were very similar (see PIP calculation method in Chapter 4). A list of chromosomes was also evolved which defined how to examine the shape contours for groups of line-segment vectors that were not similar. Each chromosome had associated with it the mean and standard deviation data for each set of line-segment vector appropriate for the set of example contours (Figure 6-1 and Figure 6-2). The set of example contours was referred to as the training set, and the mean and standard deviation data for each line-segment were referred to as the 'features' of that part of the contour examined by the instructions in the chromosome.

Ident	Chromosome	Fitness
1	**0 0 2 0.19 2**	**0.81**
2	0 0 2 0.13 2	0.81
3	0 0 2 0.13 0	0.81
4	0 0 2 0.13 1	0.81
5	0 0 2 0.10 1	0.81
6	0 0 2 0.16 2	0.81
7	0 0 2 0.16 1	0.81
8	0 0 2 0.16 0	0.81
9	0 0 2 0.19 1	0.81
10	0 0 2 0.19 0	0.81
11	0 0 2 0.03 1	0.67
12	0 0 2 0.94 0	0.67
13	0 0 2 0.00 1	0.67
14	0 0 2 0.03 0	0.67
15	0 0 2 1.00 0	0.67
16	**0 0 2 0.90 1**	**0.67**
17	0 0 2 0.00 2	0.67
18	0 0 2 0.97 1	0.67
19	0 0 2 0.97 0	0.67
20	0 0 2 0.90 0	0.67

Figure 6-1: Chromosome List – Chromosomes 1 to 20 (GA Test 1)

Generation 17

```
        Chromosome        Fitness
 1: 0  0  2  0.19  2     0.809
    Mean Data
       0.00       -- Sequence Finish Quadrant
   0.00   0.00 -- Quadrant
   9.60  32.50 -- Length
   6.00   7.00 -- Direction
Standard Deviation Data
       0.00       -- Sequence Finish Quadrant
   0.00   0.00 -- Quadrant
   4.25  11.87 -- Length
   0.00   0.00 -- Direction
```

Generation 18

```
        Chromosome        Fitness
16: 0  0  2  0.90  1     0.670
    Mean Data
       0.00       -- Sequence Finish Quadrant
   0.00   0.00 -- Quadrant
  10.00  32.90 -- Length
   5.60   6.40 -- Direction
 Standard Deviation Data
       0.00       -- Sequence Finish Quadrant
   0.00   0.00 -- Quadrant
   4.07  11.84 -- Length
   1.20   1.80 -- Direction
```

Figure 6-2: Chromosome Mean & Standard Deviation Data (GA Test 1)

Shape recognition requires some form of classification method that is able to partition the various distinguishing features of a shape and its contour, into groups or classes, such that the overlap between the classes is small. A review of related work on classification and recognition methods is provided (see Section 6.2). The results from the research described in this chapter indicated that the use of the genes in the chromosome from the genetic algorithm of Chapter 5 did enable a new and effective recognition scheme to be developed.

This research work also suggested that the threshold parameters required by this recognition process could, themselves, be the subject of evolution, using another optimising genetic algorithm. The development of the latter genetic algorithm is suggested as a suitable topic for future research. The research work described in this chapter investigated one particular method for using the output of the genetic algorithm discussed in Chapter 5. The investigation of other recognition methods using

the chromosome encoding is discussed further in Chapter 9.

6.2 Review of Related Work

References to general recognition algorithms are listed in Figure 6-3. Some of the references use a genetic algorithm to adjust the necessary parameters associated with each type of recognition or classification method.

Related Work	References
Nearest-Neighbour/Neural Network Classifier	Babu and Murty, 1993; Brill, 1992; Djouadi and Bouktache, 1997; Hemminger and Pomalaz-Raez, 1996; Ho, 1992; Hoffman, 1993; Hung and Adeli, 1994; Kuncheva, 1997; Lee, 1995; Mitzias and Mertzios, 1994; Murthy and Chowdhury, 1996; Smith, 1994; Zhao and Higuchi, 1996.
Statistical/Syntactical Recognition Techniques	Bala and Wechler, 1996; Cowell, 1995; Huang and Gu, 1993; Sonka, 1994; Tanaka, 1995; Yang, 1991.
Recognition Matching Techniques	Basak, 1995; Brunelli and Poggio, 1993; Cheng and Yan, 1998; Chen, 1997; Chen, 1994; Dufresne and Dhawan, 1995; Huang and Wang, 1996; Lo and Tsai, 1997; Pao and Li, 1992; Roh and Kweon, 1998; Wu and Sheu, 1996.
Genetic Based Machine Learning (GBML)	Dorigo and Schnepf, 1993; Gelsema, 1996; Hsu and Hwang, 1997; Lee, 1996; Lovell and Bradley, 1996; Neri and Saitta, 1996.
Tree and Graph Recognition	Abuhaiba and Ahmed, 1993; Pernus, 1994; Yang and Prasad, 1993; Zhou and Pavlidis, 1994.

Figure 6-3: Recognition Algorithm References

Classification can be defined (James, 1985) as "The assignment of an object to one of a number of predetermined groups based on observations made on that object." The same author points out that classification is not 'cluster analysis' or 'analysis of variance'. Cluster analysis checks a set of data for any tendency to form groups. Analysis of variance provides statistical proof that groups exist within the data. James (1985) also noted that analysis of variance could be used to confirm that the

classification groups are different enough to make accurate assignment possible. Good classification requires the following:

1. A classification rule, which is a well-defined procedure that can be described and applied without the need for any additional 'subjective' judgements.
2. An assessment of the performance of the 'rule'.
3. An understanding of how the rule achieves its results.

A measurement of the performance of the rule should provide the 'best' rule. The best rule is one that minimises the total error of classification or "the probability that the rule will misclassify an object is small." James (1985) developed Bayes' rule and states that the best classification is achieved when an object is assigned to the group with the highest 'conditional' probability and that this assignment minimises the total error of classification.

Jozwik (1998) compared a parallel k-NN (k-nearest neighbours) classifier with a standard version of the algorithm. Component classifiers decided between two classes only and the number of possible pairs of classes determined the number of component classifiers. A global decision was made by voting on the output of all of the component classifiers. It was shown that the replacement of the k-NN rule by a combined (1-NN, k-NN) rule reduced the computation time requirement and the parallelisation of the classifier structure decreased the error rate. The effectiveness of the approach was verified using samples that were derived from multi-sensor remote-sensing images.

Sonka (1994, page 283) distinguished between statistical pattern recognition and syntactical pattern recognition. Statistical pattern recognition is described as a quantitative description of objects using numerical parameters, while a qualitative description of an object is a characteristic of syntactic pattern recognition. A syntactic object description should be used whenever a quantitative feature description (vector) is not able to represent the complexity of the described object and/or when the object can be represented as a hierarchical structure consisting of simpler parts. The elementary

properties of the syntactically described objects are called primitives. An example of such a primitive would be the syntactic description of object borders with the border primitives representing parts of borders with a specific shape. After each primitive has been assigned a symbol, relations between primitives in the object are described, and a relational structure results. Sonka (1994) noted the following principles:

1. The number of primitives should be small.
2. The chosen primitives must be able to form an appropriate object representation.
3. Primitives should be easily segmentable from the image.
4. Primitives should be easily recognisable using some statistical pattern recognition method.
5. Primitives should correspond with significant natural elements of the described object (image) structure.

Sonka (1994) also pointed out that syntactic pattern recognition has a learning process associated with it that constructs a description grammar for each class of objects, and that the main difference between statistical and syntactic recognition is in this learning process.

Pao and Li (1992) presented a shape matching technique, also based on the straight line Hough transform. A shape signature is obtained by calculating the distances between pairs of points having the same Hough-space angle value. This is equivalent to calculating the perpendicular distances between pairs of parallel tangents to the curves. Matching two signatures then only amounts to calculating a 1D correlation. This shape matching technique assumed that a contour is characterised by its set of tangent lines. "Even though shape matching is a conceptually simple problem, its time complexity is enormous. In order to bring down the complexity, we need to decompose the high dimensional parameter space into several lower dimensional subspaces such that the parameters of individual subspaces can be determined."

Cowell (1995) discussed a syntactic pattern recogniser for vehicle number recognition and states that the syntactic approach has significant advantages in pattern representation and identification. Expressing complex patterns in terms of a small set of simple primitives with rules for describing the spatial relationship between them is very attractive. The use of syntactic pattern recognition as a formalisation of the work, by various researchers, on describing a boundary, by expressing shapes in terms of a series of primitives, each of which has a defined direction and size, is noted by this author. The major difficulty of the choice for the optimum set of primitives and the best grammar to state how the primitives are to be combined is also discussed.

Lovell and Bradley (1996) proposed a rule-based inductive learning algorithm called 'Multi-Scale Classification' and notes that statistical classifiers have major disadvantages, in that they rely on simplifying the assumptions about the probability distributions and decision surfaces which may not be valid. They also perform poorly when presented with non-linear relationships. The multi-scale algorithm, proposed in this paper, used a decision tree to store the knowledge it learns and classifies the training data by successively splitting the feature space in half.

Murthy and Chowdhury (1996) used a genetic algorithm in an attempt to optimise a specified objective function related to a clustering problem. This paper notes that cluster analysis groups a set of patterns (usually vectors in a multi-dimensional space) into clusters in such a way that patterns in the same cluster are similar in some sense. Likewise, patterns in different clusters are dissimilar. The clustering algorithm used a method which is based on the optimisation of a specified objective function that attempts to minimise the sum of the squared Euclidean distances between patterns and cluster centres. The genetic algorithm, in this paper, is used to find a global solution to this minimisation problem and uses a variable mutation probability to optimise the genetic search. Murthy and Chowdhury (1996) also noted that exhaustive enumeration could not lead to the required solution for most practical problems in a reasonable computation time. "Approximate heuristic techniques seeking a compromise or looking for a local minimum solution, which is not

necessarily global, have usually been adopted."

Tang (1998) proposed a new off-line handwritten Chinese character recognition system based on multi-feature and multi-level classification. Ten classes of multi-features were used and the multi-level classification scheme consisted of a group classifier with a five-level character classifier. The new techniques of overlap clustering and Gaussian distribution selection were described. Experimental results were presented that showed a recognition rate of about 90% for a unique candidate and 98% for multi-choice with ten candidates.

Neri and Saitta (1996) showed that there are problems for which the use of genetic algorithm based learning systems can be at least as effective as traditional symbolic or connectionist approaches. It is also noted, in this paper, that genetic algorithms seem to emerge in Machine Learning as an appealing alternative to classical search algorithms for exploring large spaces of hypotheses. The intrinsically parallel nature of the genetic algorithm, exploitable by distributed processing hardware, is of fundamental interest for machine learning, where computational complexity is a major limitation to the applicability of most available methodologies.

Bala (1997) described a hybrid methodology that integrated genetic algorithms and decision-tree learning in order to evolve useful subsets of discriminating features for recognising complex visual concepts. The genetic algorithm was used to search the space of all possible subsets of a large set of candidate discrimination features. Candidate feature subsets were evaluated using a decision-tree learning algorithm. The classification performance of the resulting decision-tree on unseen test data was used as the fitness of the underlying feature subset. Experimental results were presented to show how increasing the amount of learning significantly improved feature set evolution for difficult visual recognition problems.

Cheng and Yan (1998) proposed a character recognition method for digit characters that is based on

topological properties, positions of starting point, statistical analysis and shape recognition of unconstrained hand written digits using various contour information. The digit features are defined into three groups:

1. Normalised length of the digit contour, the normalised area of the digit and Fourier descriptors of the outer contour of the digit.
2. Relative position of the outer and interior contour centroids, the position of the interior contour centroid, the mean and variance of an outer contour distance measure and Fourier descriptors of the outer contour.
3. Digit eight and other digits that have a large variation in writing styles.

Chen (1996) described, in analytic form, a planar contour consisting of line segments and circular arcs. The contour matching is formulated as an optimisation problem based on the criterion of minimax (or minimal zone) errors. An iterative descent algorithm was developed to solve the multi-modal and non-differentiable optimisation problem, the solution of which reports the out-of-profile error and determines the scale and pose of the contour.

The research work described in this chapter to design a recognition method takes a different and novel approach to the usual classification and recognition techniques above. In conjunction with the genetic algorithm of Chapter 5, this new recognition method has parallel processing properties that could make it suitable for a real-time implementation. Possible development in the form of unsupervised learning, using a second genetic algorithm concurrently evolving recognition threshold parameters, may be possible and is recommended as a topic for further research (see Chapter 9).

6.3 Contour Shape Features

The conventional shape features, as reported in the literature, which are developed from a training set and used as input to a statistical classification system are such data items (features) as:

1. Contour Fourier descriptors.
2. Region and bounding-contour moment descriptors.
3. Contour curvature (of first differential edges and second differential zero-crossings), possibly at various scales.
4. Texture characteristics of regions that are bounded by closed shape contours.

Flusser and Suk (1993 and 1994) discussed several such features and mention Fourier descriptors, character stroke vectors and moment invariants. Affine moment invariants are derived that are able to describe variations in the slant of a character, and are effective features for the recognition of hand written characters. These features are also applied to the recognition of closed-boundary regions that are detected in satellite images.

Abu-Mostafa and Psaltis (1984) provided an *analytic* characterisation of moment invariants as features for pattern recognition and found that moment invariants are not *in general* good features. "They suffer from information loss, suppression and redundancy that limit their discrimination power. However, there are specific instances when these drawbacks do not essentially degrade the performance. When the images in question do not have significant information in the higher order coefficients, nothing is really lost, and when the central parts of the images have little useful detail, nothing is really suppressed."

The chromosomes evolved by the genetic algorithm in the research discussed in Chapter 5 can now become another part of the feature database of a shape contour. A selection of chromosomes that have the best recognition capabilities (not necessarily with the largest fitness values) should be used (see example chromosomes in Figure 6-1). These chromosomes were evolved during GA Test 1 (see Chapter 5). Figure 6-4 also lists a further 10 chromosomes.

Ident	Chromosome					Fitness
21	0	0	2	0.00	0	0.67
22	0	0	2	0.23	3	0.65
23	0	0	2	0.23	1	0.64
24	0	0	2	0.23	0	0.64
25	0	0	2	0.58	1	0.64
26	**0**	**0**	**6**	**0.13**	**0**	**0.63**
27	0	0	6	0.16	0	0.63
28	0	0	6	0.19	1	0.63
29	0	0	6	0.19	0	0.63
30	**0**	**0**	**6**	**0.13**	**1**	**0.63**

Figure 6-4: Chromosome List – Chromosomes 21 to 30 (GA Test 1)

Each chromosome has associated with it the mean and standard deviation values for each of the appropriate contour line-segments from the training set (see examples in Figure 6-2). The mean and standard deviation for the two line-segment vectors specified by the observation genes 1, 2 and 3 that have the value 0-0-2 are shown in Figure 6-2.

The features of the specified line-segment vectors are:

1. Finish quadrant for the last line-segment.
2. Finish quadrant for each line-segment.
3. Line-segment length.
4. Line-segment direction (0 to 7).

The number of PIPs associated with each contour in the training set could be included in the chromosome database, but was not used in the new recognition technique described below. The mean and standard deviation of the number of PIPs on a contour could be used as added extra evidence for the recognition process.

Rather than use the statistical pattern recognition methods, described in the review of related work above, for shape classification and recognition, a new and effective recognition method was developed during the research work. This new method made use of the evolved chromosomes from a training set to specify how to examine a test contour presented from the external environment. This new recognition technique, see Section 6.4, is similar to the syntactic pattern recognition

methods discussed in Sonka (1994) and is associated with a learning process that is absent in statistical pattern recognition. Olstad (1991) and Olstad and Torp (1996) also discussed a method for the use of active contours with a grammatical representation for shape recognition. The grammatical descriptions, discussed in these papers, are used to model the *a priori* knowledge available about the objects that the algorithm should recognise. One of the main uses of this type of syntactic information, as described in these papers, was to modify the energy function of the active contour locally according to the grammatical interpretation.

The new recognition method calculated a fitness value for the line-segment vectors, as specified by a training set chromosome, from the input test contour (produced by the PIP process in Chapter 4) using the same triangular fitness function that was used by the genetic algorithm in Chapter 5. The mean and standard deviation data from the training set were used to define the parameters of this triangular fitness function. The fitness of the line-segment vectors in the test set was, therefore, calculated in exactly the same way as in the fitness function used by the genetic algorithm of Chapter 5 for the line-segment vectors in the training set.

6.4 Description of Recognition Experiments

The training set used in the previous chapter is shown in Figure 6-5. The upper row of the digit two examples are from the Essex University database of digitised images of hand-written postcodes taken from envelopes as used by the Open University T396 Block 3 Technology third level course entitled 'Artificial Intelligence for Technology'. The lower row is from a digit two set, hand written by the author. The shape name appears below each example together with a numeric identification. The number of line-segments and the aspect ratio of each example are also provided. The test sets A and B are shown in Figure 6-6 and Figure 6-7.

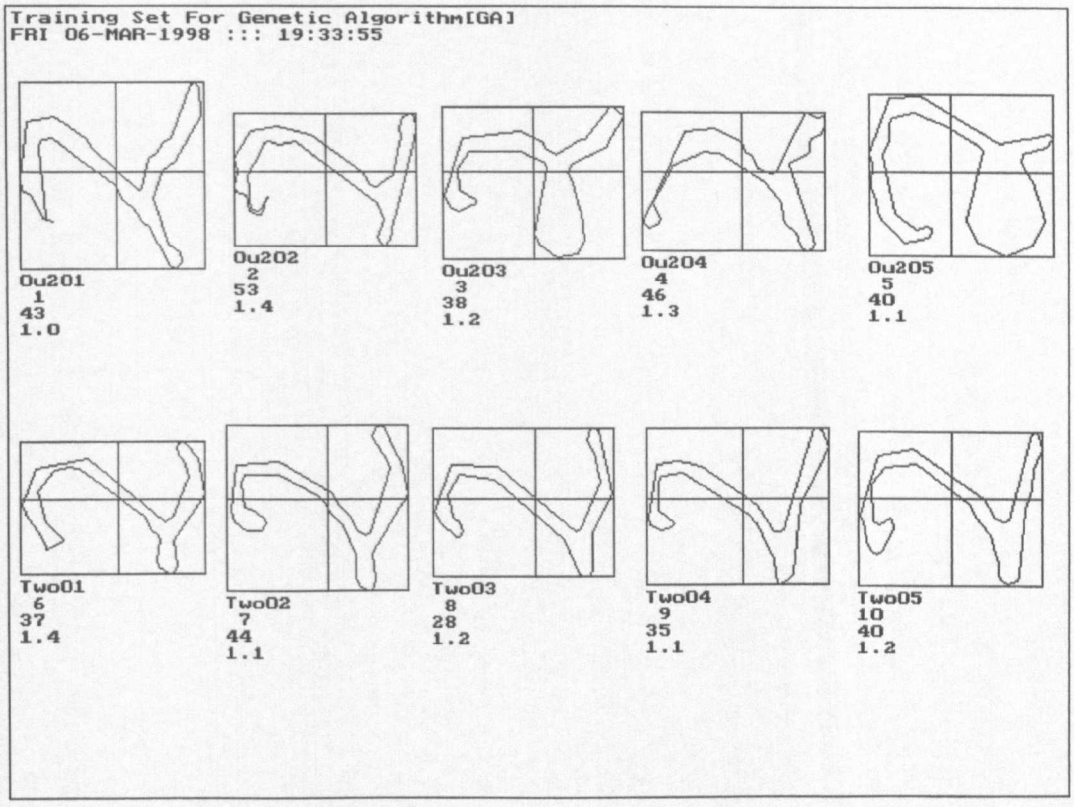

Figure 6-5: Training Set: Digit Two

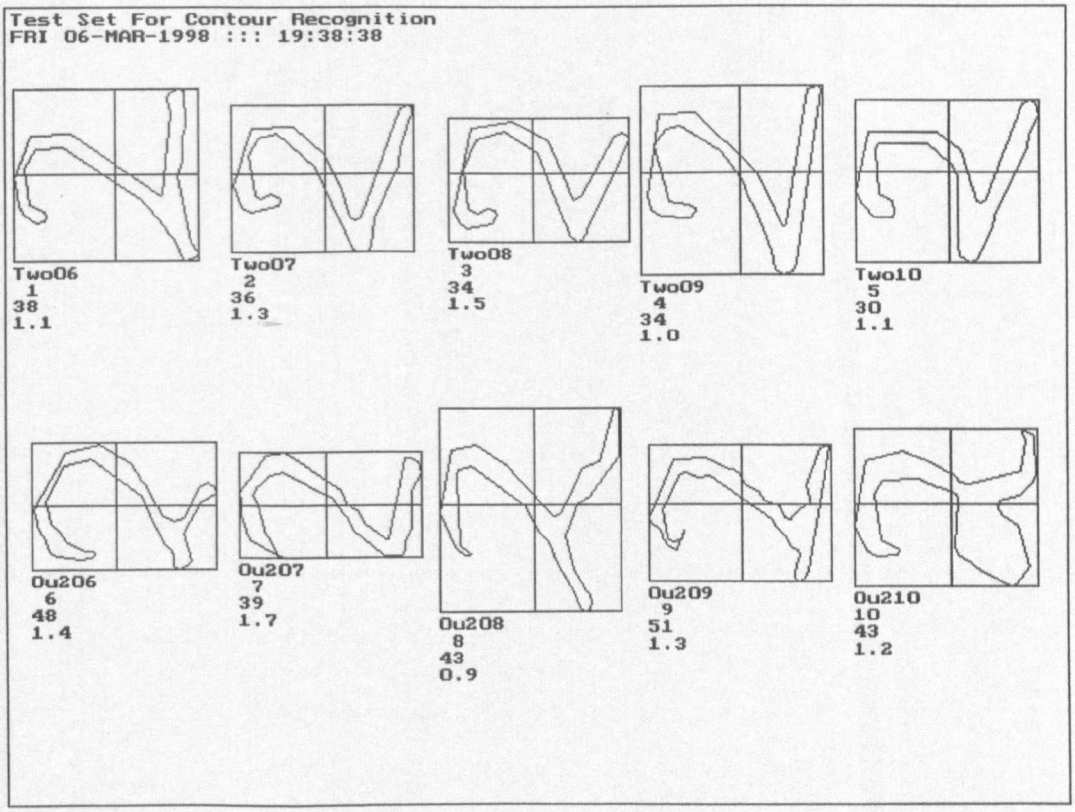

Figure 6-6: Test Set A

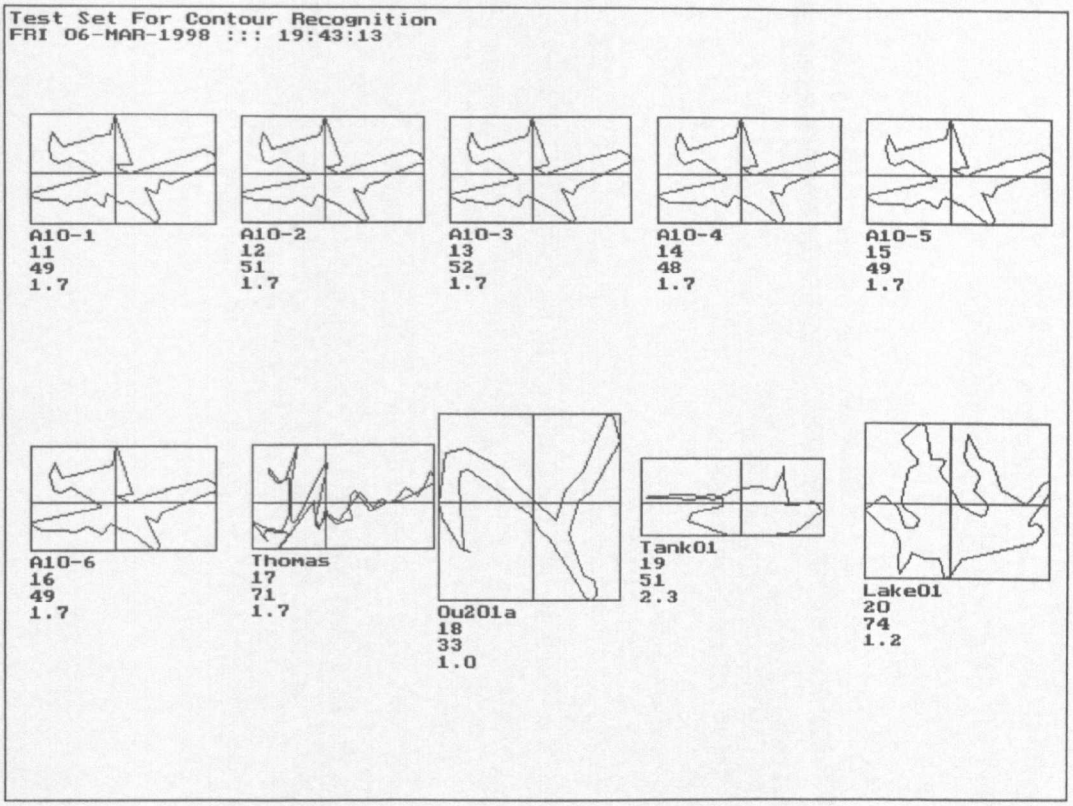

Figure 6-7: Test Set B

The test set consisted of the following examples:

1. More examples of the training set data.
2. A clipart aircraft with different sub-sampling (from 64 to 2048 points per contour) prior to normalisation.
3. A sample of the author's signature.
4. A digit two as used in the training set but with a larger sampling window on the PIP construction (which produces less PIP line-segments).
5. A clipart tank and a scanned lake contour from Held (1994).

A list of the twenty fittest solutions for a mutation probability of 0.01 and a crossover probability of 0.6 (see GA Test 1, Chapter 5) is shown in Figure 6-1, with the chromosomes 1 and 16 highlighted. The chromosome genes are shown in order, with the chromosome fitness in the right hand column. The chromosome line-segment vector mean and standard deviation data from the training set for these highlighted chromosomes is shown in Figure 6-2. Chromosome number 1 was evolved on

generation 17 and chromosome number 16 was evolved on generation 18. The mean data is shown in the upper set of numbers and the standard deviation data is shown in the lower set of numbers. The data refers to the finish quadrant for the sequence of line-segment vectors, the start quadrant for each line-segment (two in this example), the length of each line-segment and its direction from the chain codes 0 to 7 (see also Chapter 5).

The fitness for the line-segment vectors of each test set contour was calculated for each chromosome using the triangular fitness function. A good/bad score, for each test contour line-segment combination, was calculated that shows how many times the fitness is greater than a specified recognition threshold (Rthreshold). If g (good) indicates the score for a fitness value greater than or equal to Rthreshold and b (bad) indicates a score for a fitness value less than Rthreshold, we have:

$$\text{If (fitness)} \geq \text{Rthreshold then } g = g + 1 \text{ else } b = b + 1 \tag{6.1}$$

A recognition measure (R) is then calculated as:

$$R = (g - b)/(g + b) \tag{6.2}$$

Figure 6-8 shows the recognition matrix for the twenty fittest chromosomes. The value of Rthreshold for these recognition results is set to the value 0.25. The test objects are listed vertically and the score for each chromosome is shown horizontally. A one indicates a fitness value for the line-segments on the test contour greater than or equal to 0.25 and a zero indicates a fitness value, for the same line-segments, less than 0.25. The three columns on the far right of the table show the total number of ones and zeros, and the value of R, respectively.

	Ident	Chromosome																				g	b	R
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	**16**	17	18	19	20			
1	Two06	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
2	Two07	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
3	Two08	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
4	Two09	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
5	Two10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
6	Ou206	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
7	Ou207	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
8	Ou208	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
9	Ou209	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	**1**	1	1	1	1	17	3	0.70
10	Ou210	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
11	**A10-1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**20**	**0**	**1.00**
12	A10-2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**1**	1	1	1	1	10	10	0.00
13	A10-3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**1**	1	1	1	1	10	10	0.00
14	A10-4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**1**	1	1	1	1	10	10	0.00
15	**A10-5**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**20**	**0**	**1.00**
16	A10-6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**1**	1	1	1	1	10	10	0.00
17	Thomas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	20	-1.00
18	Ou201a	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	20	0	1.00
19	Tank01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	20	-1.00
20	Lake01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	20	-1.00

Figure 6-8: Recognition Matrix – Chromosomes 1 to 20 (GA Test 1)

6.5 Discussion of Recognition Results

The recognition matrix is shown in Figure 6-8 for the digit two, with scores of 20/0 and 17/3, indicating that the twenty fittest chromosomes are able to recognise the digit two from the test set. Parts of the clipart aircraft also show high fitness for these chromosomes and two incorrect recognition scores (false alarms) are therefore indicated. The false alarm contours 11 and 15 (Ident A10-1 and A10-5) are highlighted in Figure 6-8. The individual fitness for each contour, in test sets A and B, for the chromosomes 1 and 16 (Figure 6-1) is shown in Figure 6-9 and Figure 6-10. Note rows 1 to 4 show contours 1 to 5, 6 to 10, 11 to 15 and 16 to 20 respectively. Also note the high fitness for contours 11 and 15.

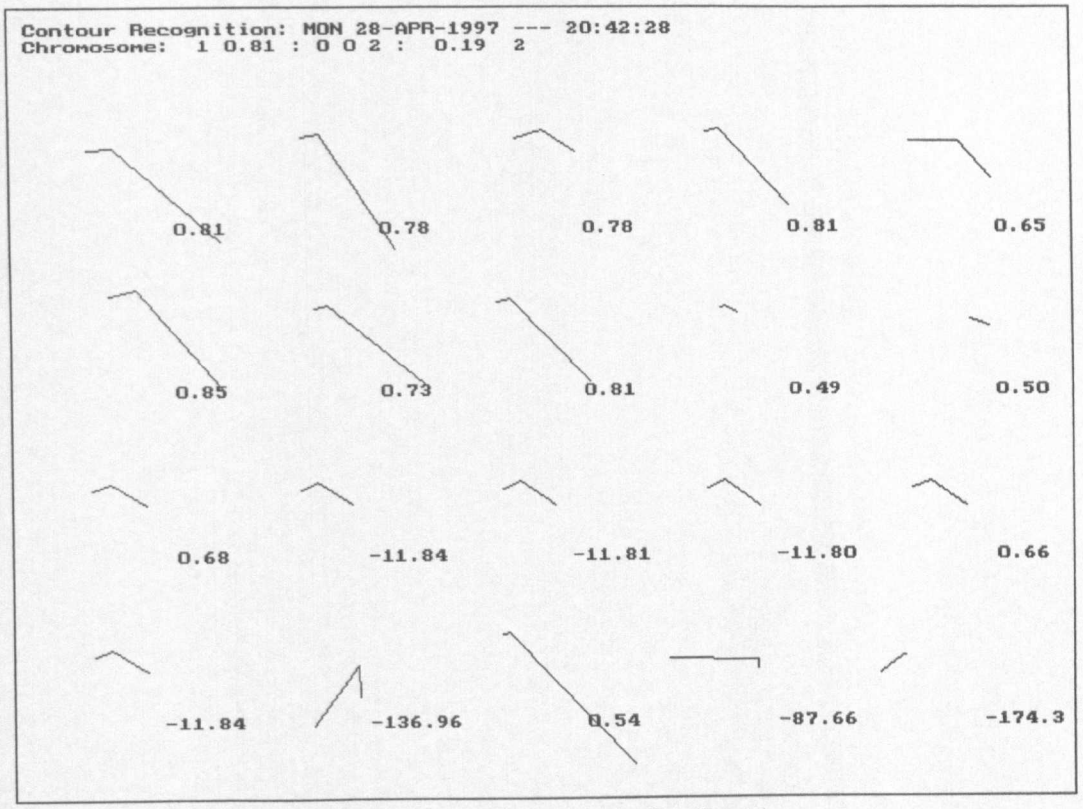

Figure 6-9: Recognition Using Best Chromosome 1: GA Test 1

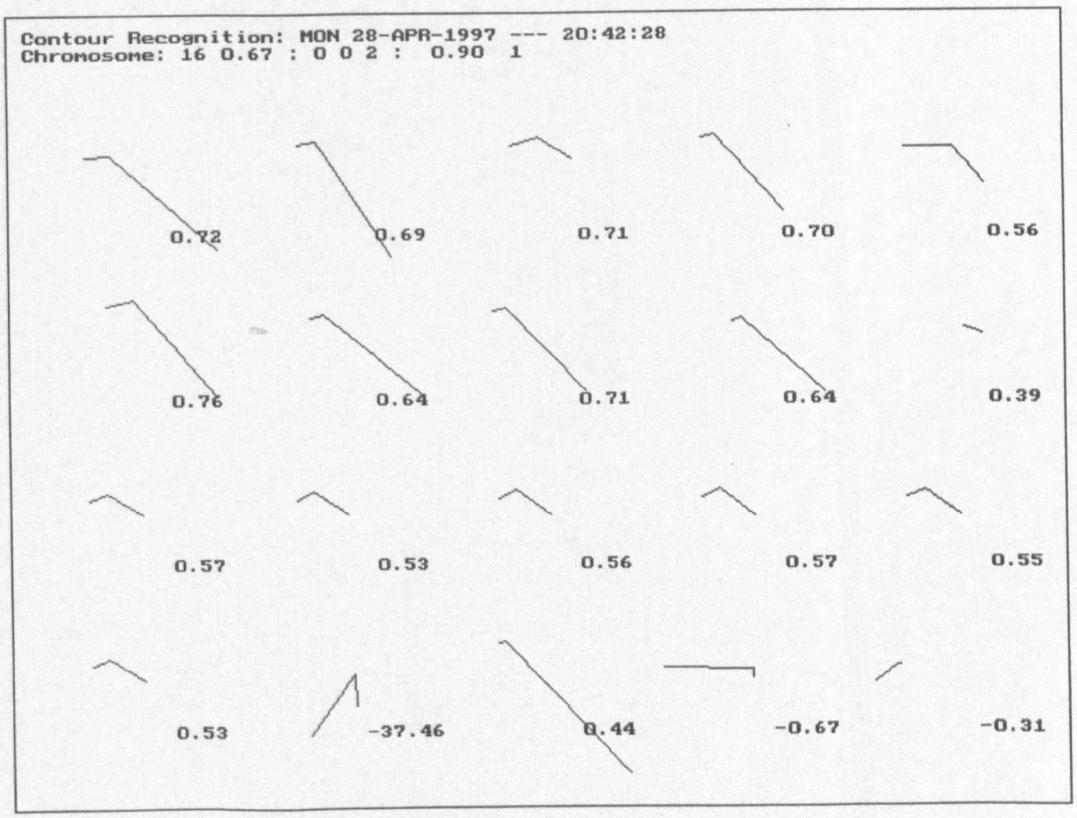

Figure 6-10: Recognition Using Chromosome 16: GA Test 1

Chromosomes 21 to 30 are shown in Figure 6-4 (GA Test 1) with the chromosomes 26 and 30 highlighted. The recognition matrix for these chromosomes, Figure 6-11, shows:

1. Failure to recognise contour 9 (OU209 - a digit two contour).
2. Zero false alarms.
3. 100% recognition, with no false alarms, for chromosomes 26 and 30.

	Ident	Chromosome	g	b	R
		21 22 23 24 25 **26** 27 28 29 **30**			
1	Two06	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
2	Two07	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
3	Two08	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
4	Two09	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
5	Two10	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
6	Ou206	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
7	Ou207	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
8	Ou208	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
9	**Ou209**	**1 0 0 0 1 1 0 0 0 1**	**4**	**6**	**-0.2**
10	Ou210	1 1 1 1 1 **1** 1 1 1 **1**	10	0	1.0
11	A10-1	1 1 1 1 0 **0** 0 0 0 **0**	4	6	-0.2
12	A10-2	1 0 1 1 0 **0** 0 0 0 **0**	3	7	-0.4
13	A10-3	1 1 1 1 1 **0** 0 0 0 **0**	5	5	0.0
14	A10-4	1 1 1 1 0 **0** 0 0 0 **0**	4	6	-0.2
15	A10-5	1 1 1 1 1 **0** 0 0 0 **0**	5	5	0.0
16	A10-6	1 0 1 1 0 **0** 0 0 0 **0**	3	7	-0.4
17	Thomas	0 0 0 0 0 **0** 0 0 0 **0**	0	10	-1.0
18	Ou201a	1 0 1 1 1 **1** 1 1 1 **1**	9	1	0.8
19	Tank01	0 0 0 0 0 **0** 0 0 0 **0**	0	10	-1.0
20	Lake01	0 0 0 0 1 **0** 0 0 0 **0**	1	9	-0.8

Figure 6-11: Recognition Matrix – Chromosomes 21 to 30 (GA Test 1)

The contour line-segment vectors involved in the recognition process for chromosomes 26 and 30 are shown in Figure 6-12 and Figure 6-13. The high recognition measures, (> +0.33), for the contours 1 to 10 and 18, which are all line-segment vectors from the digit two contours, can be observed. The very low recognition measures, (< -33.0), for the other contours can also be seen. Note that the chromosomes 26 and 30 only differ in the threshold on the length of the line-segment vectors to be used and thus specify very similar ways to observe the contour for recognition purposes.

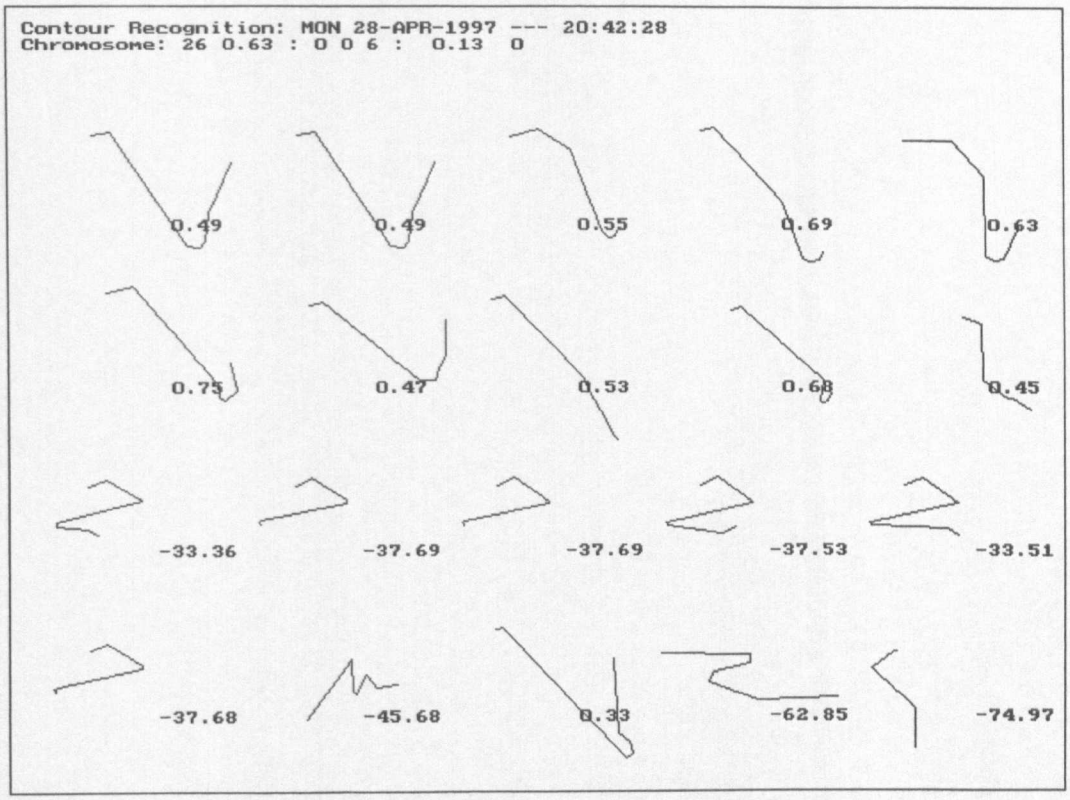

Figure 6-12: Recognition Using Chromosome 26: GA Test 1

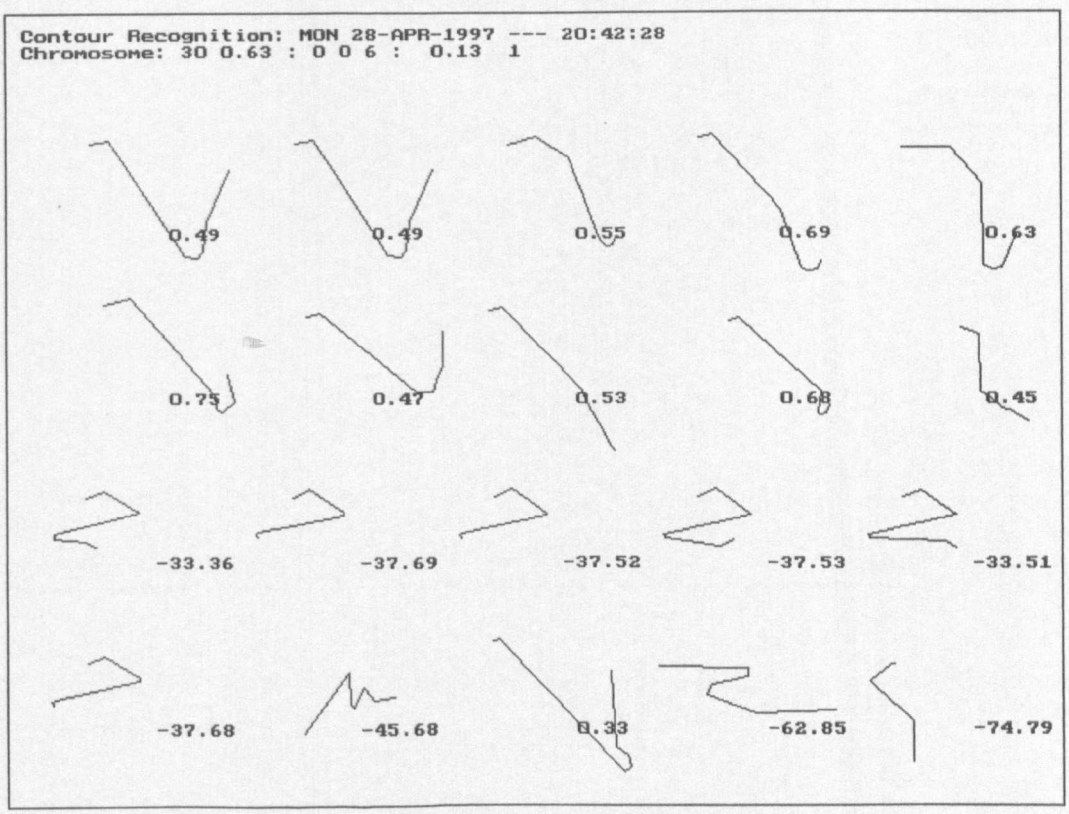

Figure 6-13: Recognition Using Chromosome 30: GA Test 1

A list of twelve chromosomes that define how to examine the shape contours for line-segments that are not similar is shown in Figure 6-14, i.e. a mismatch of the contours. These twelve chromosomes were developed by the genetic algorithm in Chapter 5 for a mutation probability of 0.5 and a crossover probability of 0.6. The genetic algorithm evolved these twelve solutions using a fitness value equal to (1.0 – measured chromosome fitness). See right-most column in Figure 6-14, where the fitness value is of the order of 0.9 indicating a measured chromosome fitness value of the order of 0.1.

Ident	Chromosome					Fitness
1	0	1	2	0.97	10	0.92
2	1	0	3	0.45	16	0.92
3	0	1	3	0.10	8	0.91
4	**3**	**0**	**4**	**0.35**	**15**	**0.91**
5	2	0	4	0.77	14	0.91
6	**3**	**0**	**4**	**0.52**	**16**	**0.90**
7	1	0	3	0.00	17	0.90
8	**2**	**0**	**8**	**0.35**	**11**	**0.90**
9	**3**	**0**	**4**	**0.84**	**15**	**0.90**
10	1	0	3	0.16	8	0.90
11	3	1	2	1.00	6	0.90
12	**0**	**1**	**6**	**0.84**	**10**	**0.90**

Figure 6-14: Chromosome List [MisMatch] – Chromosomes 1 to 12 (GA Test 43)

The recognition matrix of the sections of the contour line-segments that are not similar is shown in Figure 6-15 for each of the 12 chromosomes. Chromosomes 4, 6, 8, 9 and 12 correctly identified the dissimilar line-segments. The value of R for the contour shapes 1 to 10 and 18 (the digit two) varies from –0.5 to –1.0, also indicating good recognition of the dissimilar line-segments. This evidence from chromosomes 4, 6, 8, 9 and 12 could be added to that of the chromosomes previously discussed, to reinforce the recognition of the contour shapes 1 to 10 and 18 as the digit two.

The line-segment vectors used in the recognition of the mismatched sections of the contour shapes are shown in Figure 6-16 for chromosome 6 and Figure 6-17 for chromosome 12. The mismatch between the line-segments can be seen with Figure 6-17 showing that the barrel of the tank (row 4, second from the right, contour 19) gives a value for R of –0.91. This large mismatch value of R indicates an almost total (-1.0) mismatch. This mismatch value would seem to be consistent with a

	Ident	Chromosome												g	b	R
		1	2	3	4	5	6	7	8	9	10	11	12			
1	Two06	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
2	Two07	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
3	Two08	0	1	0	0	1	0	1	0	0	0	0	0	3	9	-0.50
4	Two09	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
5	Two10	1	0	0	0	0	0	0	0	0	0	0	0	1	11	-0.83
6	Ou206	1	0	0	0	0	0	0	0	0	0	1	0	2	10	-0.67
7	Ou207	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
8	Ou208	0	1	0	0	0	0	1	0	0	1	0	0	3	9	-0.50
9	Ou209	1	0	0	0	0	0	0	0	0	0	0	0	1	11	-0.83
10	Ou210	1	1	1	0	0	0	0	0	0	0	0	0	3	9	-0.50
11	A10-1	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
12	A10-2	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
13	A10-3	0	0	0	0	1	0	0	0	0	0	0	0	1	11	-0.83
14	A10-4	0	0	0	0	1	0	0	0	0	0	0	0	1	11	-0.83
15	A10-5	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
16	A10-6	0	0	0	0	1	0	0	0	0	0	0	0	1	11	-0.83
17	Thomas	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
18	Ou201a	0	1	0	0	0	0	1	0	0	0	0	0	2	10	-0.67
19	Tank01	0	0	0	0	0	0	0	0	0	0	0	0	0	12	-1.00
20	Lake01	1	0	0	0	0	0	0	0	0	0	0	0	1	11	-0.83

Figure 6-15: Recognition Matrix [MisMatch] – Chromosomes 1 to 12 (GA Test 43)

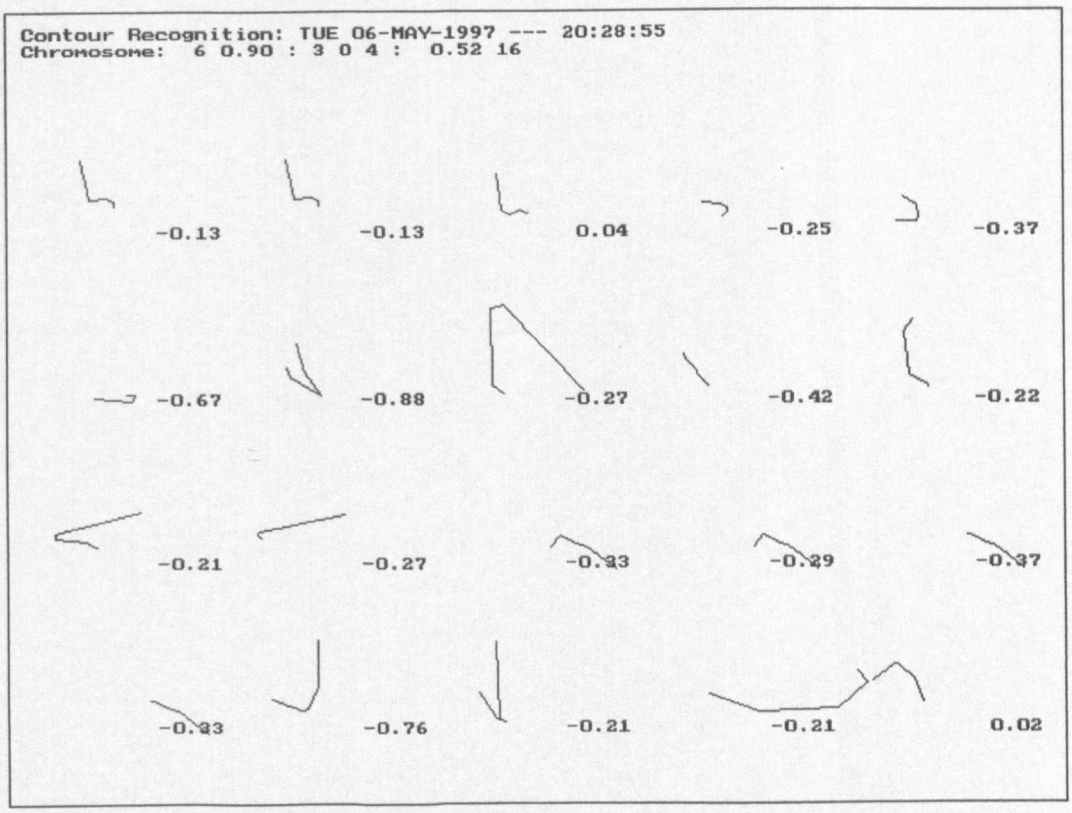

Figure 6-16: Recognition Using Chromosome 6 [MisMatch]: GA Test 43

human observer's view of that part of the contour.

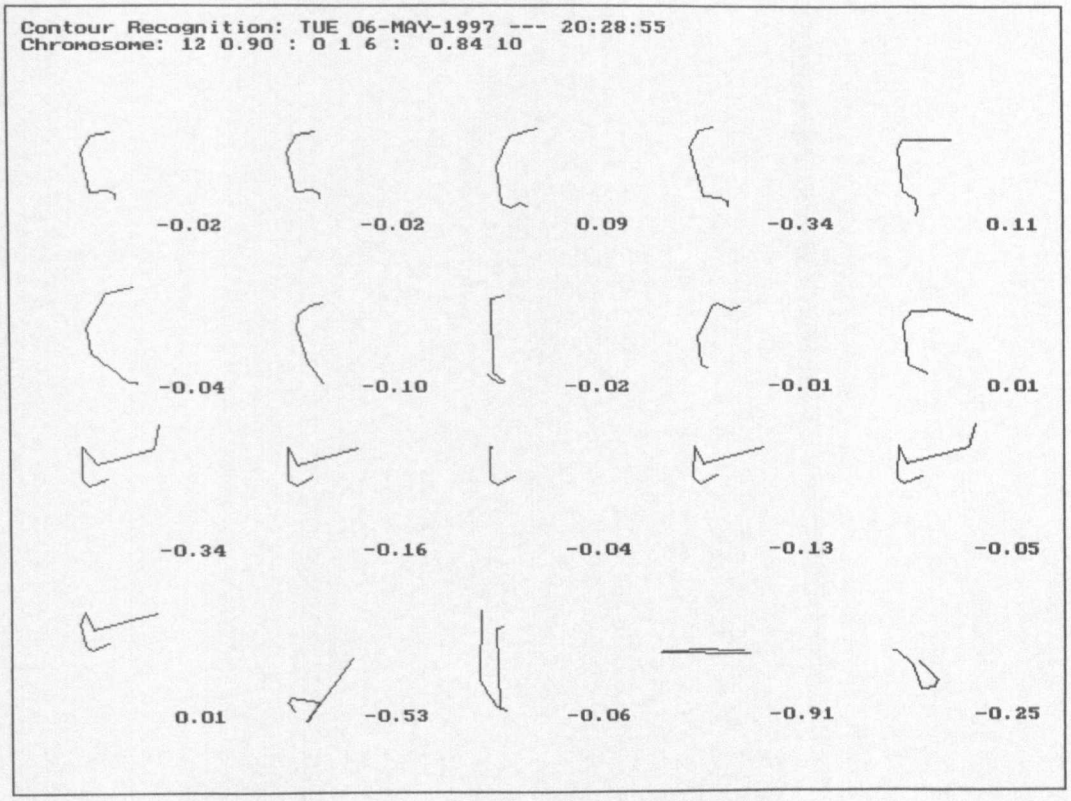

Figure 6-17: Recognition using Chromosome 12 [MisMatch]: GA Test 43

The digit four, from test set C (Figure 6-18), was added to the test set, substituting for the digit two test contours, numbers 1 to 5. The recognition matrix for this new test set C is shown in Figure 6-19 for chromosomes 1 to 16, and Figure 6-20 for chromosomes 11 to 30. Chromosome 1 recognises that the digit four (Test set numbers 1 to 5) is not the same as the digit two (Test set numbers 6 to 10 and 18). Chromosome 16 incorrectly identifies one of the digit four contours (Test set number 3).

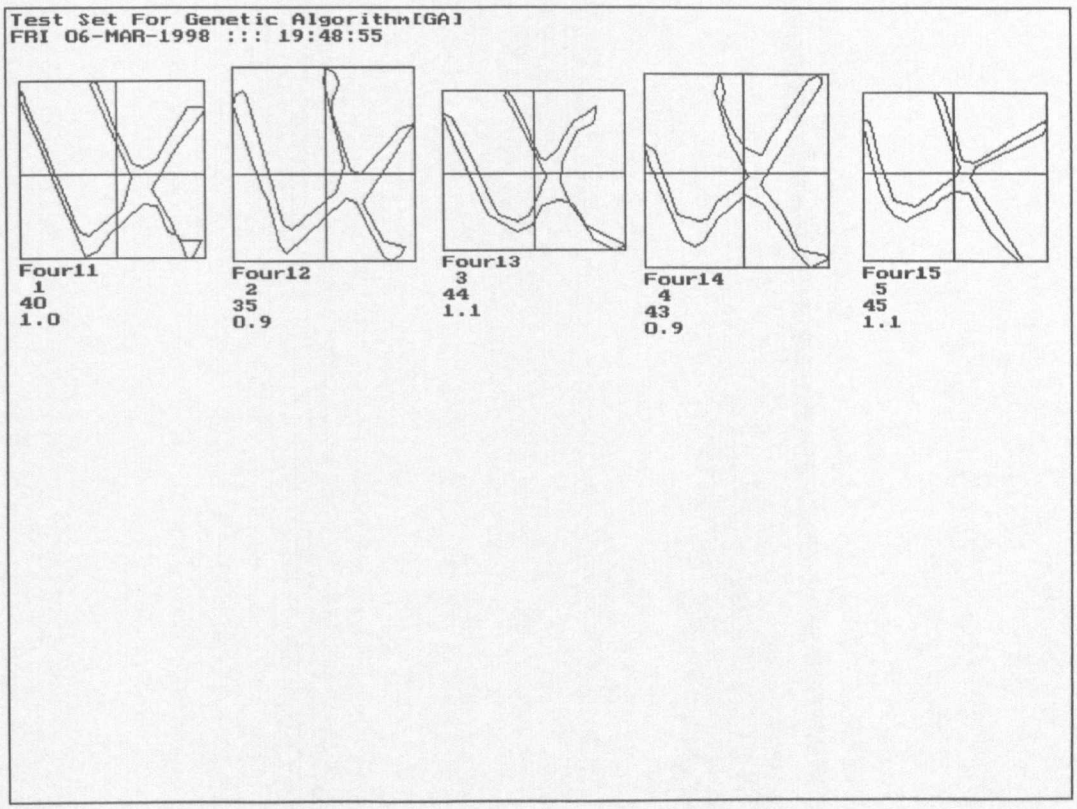

Figure 6-18: Test Set C: Digit Four

	Ident	Chromosome																g	b	R
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
1	Four11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00
2	Four12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00
3	Four13	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	6	10	-0.25
4	Four14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00
5	Four15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00
6	Ou206	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
7	Ou207	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
8	Ou208	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
9	Ou209	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	13	3	0.63
10	Ou210	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
11	A10-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
12	A10-2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	6	10	-0.25
13	A10-3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	6	10	-0.25
14	A10-4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	6	10	-0.25
15	A10-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
16	A10-6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	6	10	-0.25
17	Thomas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00
18	Ou201a	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	0	1.00
19	Tank01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00
20	Lake01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	-1.00

Figure 6-19: Recognition Matrix with Test Set C – Chromosomes 1 to 16 (GA Test 1)

	Ident	Chromosome																				g	b	R
		11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
1	Four11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-1.0
2	Four12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-1.0
3	Four13	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	11	9	0.1
4	Four14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-1.0
5	Four15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-1.0
6	Ou206	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20	0	1.0
7	Ou207	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20	0	1.0
8	Ou208	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20	0	1.0
9	Ou209	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1	14	6	0.4
10	Ou210	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20	0	1.0
11	A10-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	14	6	0.4
12	A10-2	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	13	7	0.3
13	A10-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	15	5	0.5
14	A10-4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	14	6	0.4
15	A10-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	15	5	0.5
16	A10-6	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	13	7	0.3
17	Thomas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-1.0
18	Ou201a	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	19	1	0.9
19	Tank01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-1.0
20	Lake01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	19	-0.9

Figure 6-20: Recognition Matrix with Test Set C – Chromosomes 11 to 30 (GA Test 1)

Chromosomes 26 and 30 (highlighted) correctly identify the digit two contours and also correctly identify that the digit four contours are not similar to the digit two contours. The line-segments involved in the recognition for the chromosomes 1, 16, 26 and 30 using the test set C are shown in Figure 6-21 to Figure 6-24.

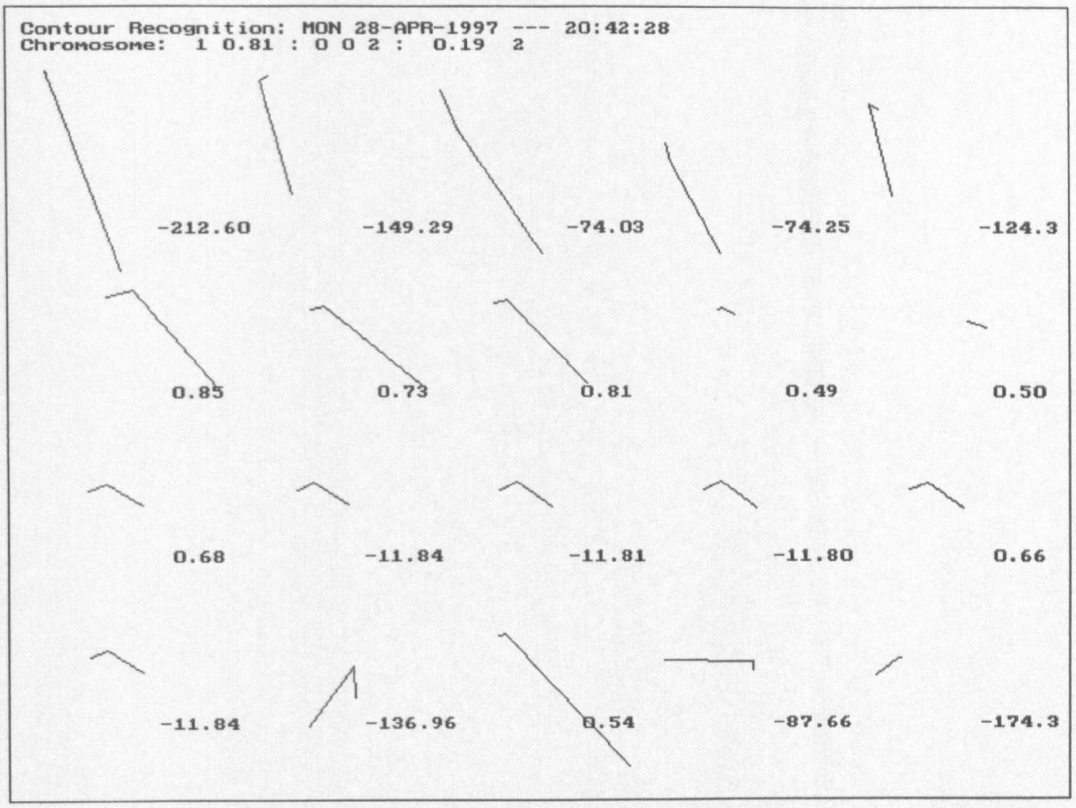

Figure 6-21: Recognition Using Best Chromosome 1 Including Test Set C: GA Test 1

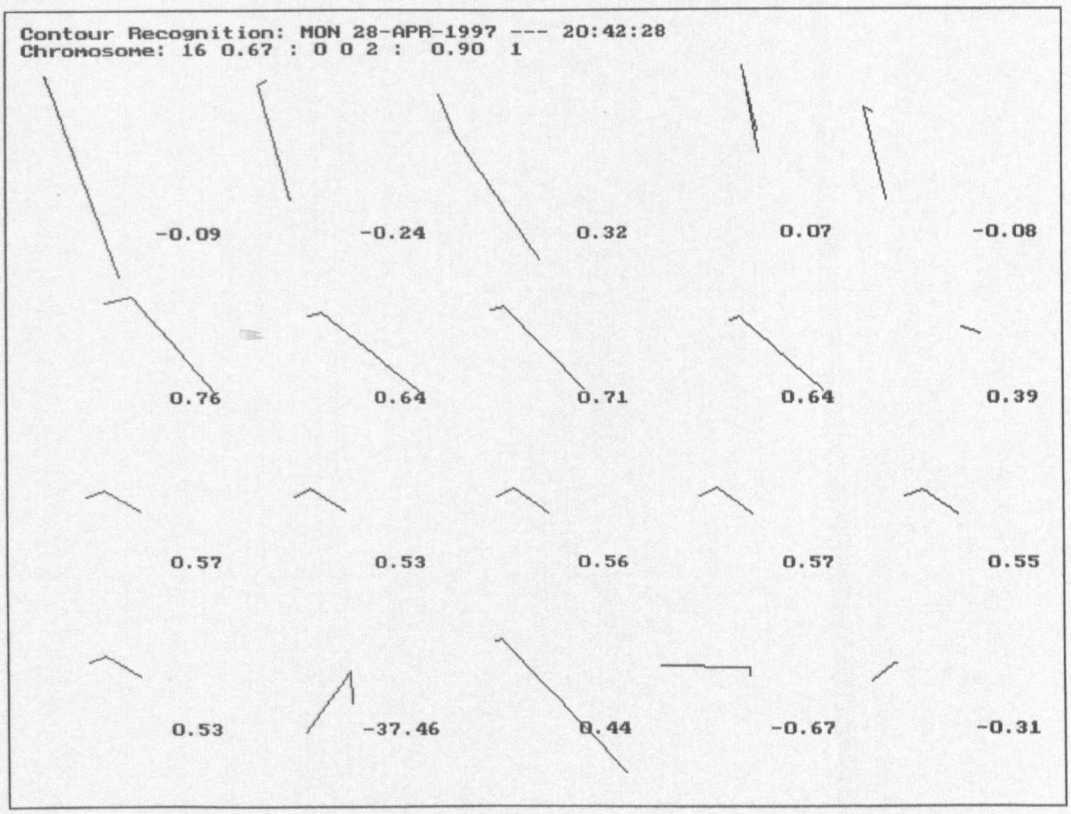

Figure 6-22: Recognition Using Chromosome 16 Including Test Set C: GA Test 1

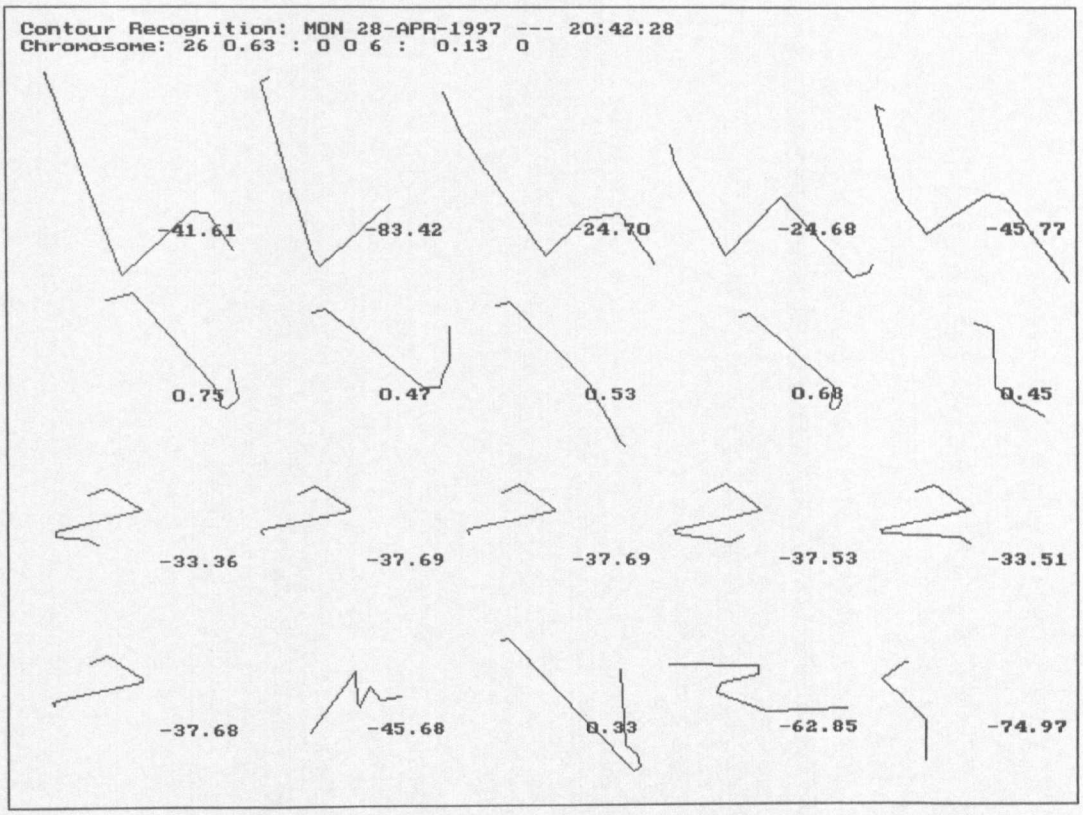

Figure 6-23: Recognition Using Chromosome 26 Including Test Set C: GA Test 1

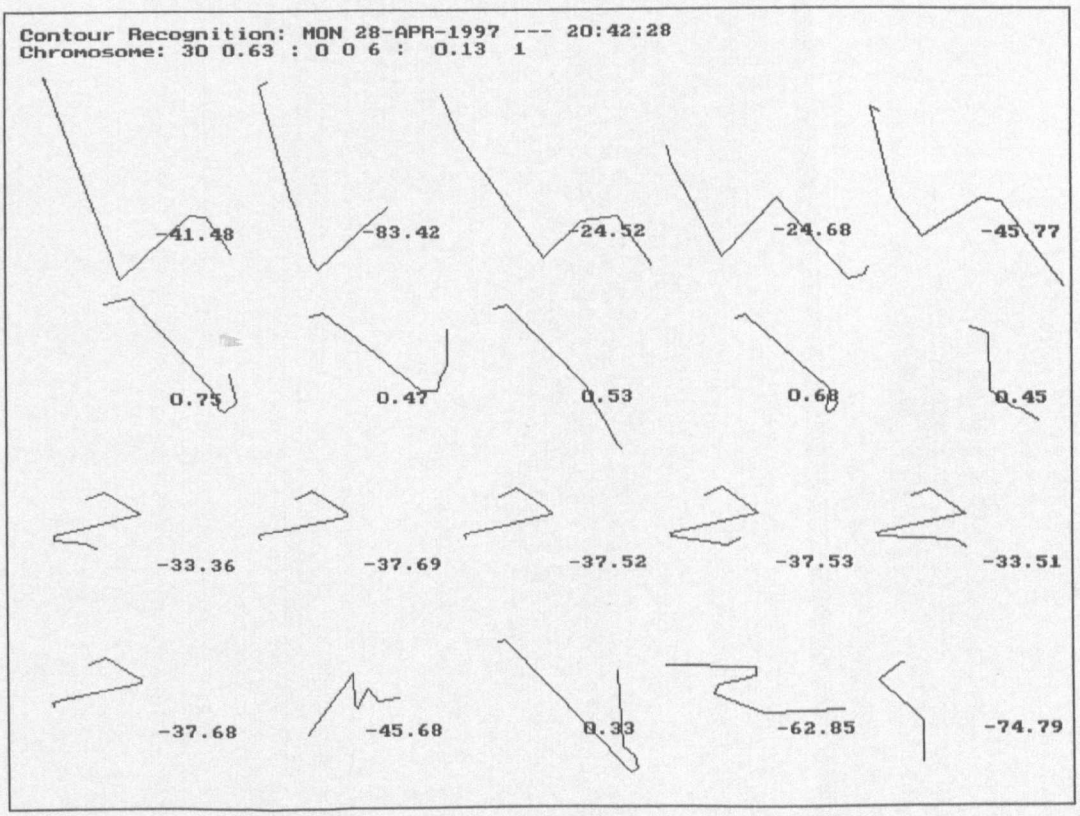

Figure 6-24: Recognition Using Chromosome 30 Including Test Set C: GA Test 1

A summary of the recognition results for the forty-four training set experiments in Chapter 5 is shown in Figure 6-25. The left-hand side of the table is shown for reference and is a summary of the behaviour of the genetic algorithm for the various values of crossover and mutation probabilities. The chromosome type column is indicated as 'B' for a binary type and 'I' for the integer type of chromosome. The total recognition column (Total Recn) shows the range of the number of chromosomes that correctly identified the digit two with no false alarms. The recognition (Recn) and false alarm percentage ranges were calculated for the fittest chromosomes that evolved for each of the crossover and mutation probabilities. The recognition measure R column (seeFigure 6-25 and equation (6.2)) shows the various values for R used to calculate the percentage range recognition scores.

Tests 1 to 20 each varies the mutation probability with the crossover probability equal to 0.6. For low values of mutation probability (< 0.1) a large number of solutions is obtained. The number of different solutions and diversity are low. The influence of the crossover operator dominates the selection of a few high fitness chromosomes. The fittest solutions are also very similar hence the low diversity in the population. Chromosomes with total recognition capability are evolved and the recognition % range in some cases is less than 100%. The lowest mutation probability values also give higher false alarm % ranges.

Tests 5 to 10 each indicates that, for mutation probabilities in the range 0.1 to 0.3, the number of solutions decreases with the number of different solutions and diversity increasing. The influence of the mutation operator increases over this range of values. The fittest solutions show more variety hence the increase in the diversity of the population. Chromosomes evolve with a total recognition capability but over a smaller range than for the lower mutation probability values. The recognition % range is 100 % and the false alarm % range is lower than for Tests 1 to 4. This range of mutation probabilities appears to give more consistent recognition capability with a low false alarm % range. The number of different solutions are, overall, slightly higher for this range of mutation

probabilities.

Tests 11 to 20 each shows a decrease in the number of solutions as the mutation probability increases with the diversity generally also increasing. Low diversity is obtained in Test 19 because a mutation probability of 1.0 changes the value of each bit in the chromosome, so the population continually oscillates between two sets of chromosomes whose fitness is entirely dependent on the initial random population. For the integer chromosome the mutation operator behaves slightly differently in that the value of each integer gene (in the chromosome) is either increased or decreased with a random probability hence the diversity is maintained. The total recognition range decreases and, in general, the recognition capability also decreases with an increase in the false alarm % range. The influence of the mutation operator over this range of values appears to reduce the effects of the crossover operator and hence the selection process becomes more random.

Tests 21 to 22 each shows the results for a completely random selection of the chromosomes. The results are similar to those for the high values of mutation probability. The number of solutions is low and the diversity is high. The number of chromosomes with total recognition capability is also low and the false alarm % range is high. These results appear to show that high mutation probability values produce the same effect as random generation of the chromosomes and that the action of the crossover operator is required to evolve a number of solutions that provide good recognition capabilities with a low false alarm % range.

Tests 23 to 32 each shows the effect of the mutation operator alone. The diversity of the population increases with increasing value of the mutation probability. Test 31 also gives similar diversity values to those produced in Test 19 (100% bit swapping). The number of chromosomes providing total recognition is low and, in general, the recognition capabilities are reduced and the false alarm % range increases as the mutation probability increases. These results again appear to indicate that the use of the crossover operator is essential to evolve chromosomes with 100% recognition

capabilities.

Tests 33 to 42 each shows the effect of the crossover operator alone. The crossover process dominates the selection process producing a low diversity in the population. The number of different solutions is lower than in the cases where the mutation operator was used. The recognition capability of the evolved chromosomes is also reduced with a large increase in the false alarm % range. The crossover operator is providing too much exploitation with very little exploration of the solution space. The mutation operator is therefore required to improve this exploration, as shown in Tests 1 to 20.

Tests 43 to 44 each shows the capability of the genetic algorithm in Chapter 5 to identify the sections along the training set contours (for the digit two) that are mismatched, i.e. dissimilar. The number of chromosomes providing total recognition of the mismatched contour sections is higher than for those providing a matched recognition. These results appear to indicate that the training set of digit two contours has up to eight sections on the contours that show a mismatch. The recognition capability of the chromosomes to identify the mismatched sections of the contours appears to be similar to that of the chromosomes with a matching capability (Tests 13 and 14). Identification of the contour sections that differ could be used as extra information for the recognition process.

The recognition measure (R) appears to be in the range 4 ± 2 for Tests 1 to 42. The mismatch (Tests 43 and 44) settings were negative and of a similar range but lower absolute values. The value of the recognition measure required for a good recognition capability would, therefore, seem not to be critical and is similar for the various values of the mutation and crossover probabilities.

The performance of the binary form of the chromosome appears to be similar to and, in general, slightly more consistent than the integer version of the chromosome. The mutation operator for the two types of chromosome behaves differently, with the integer chromosome providing more

diversity for mutation probabilities equal to 1.0. The use of the binary chromosome is preferred because the schema theorem of J. H. Holland can be used to analyse its performance.

The results of the training set [Digit Two] recognition tests (Figure 6-25) can be summarised as follows:

1. The most consistent recognition results (including 100% recognition) are obtained with the crossover probability equal to 0.6 and the mutation probability in the range 0.1 to 0.3. A number of different chromosomes, i.e. 'ways of observing the shape contour line-segments', are evolved using these genetic operator values. A value of 0.6 for the crossover operator is often recommended in the literature.
2. Mutation probability values < 0.1 and a crossover probability of 0.6 produce low diversity in the population and a reduced number of different chromosomes. Mutation probability values > 0.3 with a crossover probability of 0.6 produce solutions in a similar manner to a random selection of the chromosomes. In both cases the false alarm % range is increased.
3. Fewer total recognition chromosomes are produced for crossover probability equal to 0.0 and the mutation probability is varied between 0.1 and 1.0. The false alarm % range under these conditions is also higher.
4. A low diversity of the population is obtained when the mutation probability is set to 0.0 and the crossover probability is varied between 0.1 and 1.0. Under these conditions the crossover operation dominates the evolutionary process. Fewer total (100%) recognition chromosomes are produced and the false alarm % range increases.
5. A recognition capability to identify the mismatch (dissimilarity) between the contours in a training set is achieved and provides a similar consistency to that of the recognition chromosomes. This mismatch capability can be used to identify the sections of the training set contours that are dissimilar. This extra evidence can then be used to reinforce the evidence from the recognition chromosomes.

GA Test	Chromo. Type	Pm	Pc	Unique Solns.	Diff. Solns.	Diversity	Total Recn. (Range)	Recn. (% Range)	False Alarms (% Range)	R
1	B	0.01	0.6	32 ± 16	3 ± 2	6.3 ± 0.7	0 : 4	90 : 100	0 : 33	0.4 ± 0.2
2	I	0.01	0.6	9 ± 6	3 ± 2	7.6 ± 0.6	0 : 1	90 : 100	0 : 22	0.4 ± 0.3
3	B	0.05	0.6	67 ± 9	6 ± 1	3.6 ± 1.1	2 : 8	100 : 100	0 : 11	0.6 ± 0.2
4	I	0.05	0.6	11 ± 8	2 ± 2	4.9 ± 0.5	0 : 4	90 : 100	0 : 22	0.4 ± 0.2
5	B	0.10	0.6	54 ± 18	6 ± 2	1.6 ± 0.3	1 : 6	100 : 100	0 : 11	0.5 ± 0.2
6	I	0.10	0.6	19 ± 9	4 ± 2	3.2 ± 0.5	0 : 2	100 : 100	0 : 22	0.4 ± 0.1
7	B	0.20	0.6	12 ± 5	4 ± 2	1.2 ± .04	0 : 3	100 : 100	0 : 22	0.4 ± 0.1
8	I	0.20	0.6	20 ± 3	6 ± 2	2.3 ± 0.4	1 : 3	100 : 100	0 : 22	0.3 ± 0.2
9	B	0.30	0.6	12 ± 5	5 ± 1	1.2 ± 0.1	0 : 2	100 : 100	0 : 22	0.4 ± 0.2
10	I	0.30	0.6	28 ± 10	5 ± 2	1.6 ± 0.3	0 : 5	90 : 100	0 : 44	0.3 ± 0.2
11	B	0.40	0.6	8 ± 2	4 ± 1	1.2 ± 0.1	0 : 1	90 : 100	0 : 44	0.3 ± 0.2
12	I	0.40	0.6	20 ± 4	6 ± 1	1.4 ± 0.2	0 : 1	80 : 100	0 : 33	0.4 ± 0.1
13	B	0.50	0.6	7 ± 2	4 ± 1	1.1 ± 0.1	0 : 1	100 : 100	0 : 22	0.5 ± 0.2
14	I	0.50	0.6	17 ± 4	6 ± 1	1.3 ± 0.1	1 : 2	100 : 100	0 : 33	0.4 ± 0.2
15	B	0.60	0.6	8 ± 3	4 ± 2	1.2 ± 0.1	0 : 2	80 : 100	0 : 44	0.5 ± 0.2
16	I	0.60	0.6	14 ± 5	6 ± 2	1.3 ± 0.1	0 : 2	100 : 100	0 : 22	0.4 ± 0.2
17	B	0.80	0.6	7 ± 2	3 ± 1	1.2 ± 0.1	0 : 1	80 : 100	0 : 44	0.5 ± 0.3
18	I	0.80	0.6	10 ± 3	4 ± 2	1.3 ± 0.2	0 : 2	80 : 100	0 : 33	0.5 ± 0.2
19	B	1.00	0.6	7 ± 3	3 ± 1	6.8 ± 1.1	0 : 2	90 : 100	0 : 44	0.5 ± 0.3
20	I	1.00	0.6	9 ± 2	4 ± 1	2.0 ± 0.4	0 : 0	100 : 100	0 : 33	0.5 ± 0.2
21	B	N/A	N/A	8 ± 3	4 ± 1	1.1 ± 0.2	0 : 1	100 : 100	0 : 22	0.4 ± 0.2
22	I	N/A	N/A	6 ± 1	3 ± 1	1.2 ± 0.1	0 : 1	80 : 100	0 : 55	0.4 ± 0.3
23	B	0.10	0.0	20 ± 3	4 ± 2	1.9 ± 0.2	0 : 2	80 : 100	0 : 44	0.5 ± 0.1
24	I	0.10	0.0	6 ± 3	2 ± 2	3.8 ± 1.1	0 : 2	90 : 100	0 : 22	0.3 ± 0.2
25	B	0.25	0.0	8 ± 2	4 ± 1	1.1 ± 0.1	0 : 2	80 : 100	0 : 33	0.4 ± 0.1
26	I	0.25	0.0	15 ± 4	5 ± 2	1.6 ± 0.3	0 : 2	100 : 100	0 : 22	0.2 ± 0.2
27	B	0.50	0.0	9 ± 2	5 ± 1	1.2 ± 0.2	0 : 2	80 : 100	0 : 33	0.3 ± 0.1
28	I	0.50	0.0	15 ± 5	5 ± 1	1.4 ± 0.2	0 : 2	90 : 100	0 : 22	0.5 ± 0.1
29	B	0.75	0.0	6 ± 3	3 ± 2	1.1 ± 0.1	0 : 1	80 : 100	0 : 55	0.2 ± 0.2
30	I	0.75	0.0	13 ± 3	5 ± 1	1.2 ± 0.1	0 : 3	90 : 100	0 : 22	0.4 ± 0.2
31	B	1.00	0.0	2 ± 1	2 ± 1	6.4 ± 0.7	0 : 1	80 : 100	0 : 33	0.3 ± 0.2
32	I	1.00	0.0	6 ± 3	3 ± 1	2.0 ± 0.4	0 : 2	90 : 100	0 : 33	0.4 ± 0.2
33	B	0.0	0.1	2 ± 1	1 ± 1	7.7 ± 0.2	0 : 0	90 : 100	0 : 66	0.4 ± 0.2
34	I	0.0	0.1	2 ± 1	1 ± 1	7.5 ± 1.0	0 : 0	100 : 100	0 : 77	0.3 ± 0.2
35	B	0.0	0.3	4 ± 2	1 ± 1	7.4 ± 0.6	0 : 1	80 : 100	0 : 66	0.4 ± 0.2
36	I	0.0	0.3	4 ± 2	1 ± 1	7.8 ± 0.3	0 : 3	100 : 100	0 : 66	0.4 ± 0.1
37	B	0.0	0.6	8 ± 7	2 ± 2	7.3 ± 0.6	0 : 0	90 : 100	0 : 22	0.4 ± 0.2
38	I	0.0	0.6	5 ± 1	2 ± 1	7.7 ± 0.3	0 : 1	80 : 100	0 : 22	0.4 ± 0.1
39	B	0.0	0.8	10 ± 6	3 ± 3	7.6 ± 0.8	0 : 2	90 : 100	0 : 22	0.5 ± 0.1
40	I	0.0	0.8	3 ± 1	1 ± 1	7.4 ± 0.8	0 : 0	80 : 100	0 : 66	0.4 ± 0.1
41	B	0.0	1.0	12 ± 7	3 ± 1	7.9 ± 0.1	0 : 2	80 : 100	0 : 44	0.4 ± 0.2
42	I	0.0	1.0	6 ± 3	2 ± 1	7.1 ± 0.9	0 : 2	90 : 100	0 : 22	0.4 ± 0.3
43	B	0.5	0.6	21 ± 1	12 ± 3	1.1 ± 0.1	4 : 8	90 : 100	0 : 9	-.2 ± 0.2
44	I	0.5	0.6	25 ± 5	14 ± 3	1.3 ± 0.1	4 : 9	90 : 100	0 : 27	-.3 ± 0.1

Figure 6-25: Training Set [Digit Two] Recognition Summary

GA Test	Figure Ident	Start Quadrant	Anti-Clockwise/ Clockwise	Number of Segments	Search Start	Length Threshold	Fitness
3	291932	0	0	2	0.55	0	0.64
8	61620	0	0	5	0.13	1	0.63
8	61620	0	0	5	0.13	0	0.63
8	61620	0	0	5	0.13	2	0.63
8	61620	0	0	6	0.13	0	0.63
10	61644	0	0	2	0.55	4	0.64
11	61657	0	0	2	0.52	4	0.64
11	61657	0	0	2	0.52	5	0.64
13	61742	0	0	6	0.13	1	0.63
15	61807	0	0	5	0.10	1	0.63
15	61807	0	0	6	0.10	0	0.63
16	61950	0	0	2	0.55	5	0.64
17	71720	0	0	5	0.10	0	0.63
17	71720	0	0	6	0.10	1	0.63
18	71731	0	0	5	0.10	1	0.63
22	301756	0	0	6	0.10	1	0.63
23	301943	0	0	2	0.55	5	0.64
23	301943	0	0	2	0.55	4	0.64
24	301955	0	0	5	0.10	1	0.63
24	301955	0	0	6	0.10	1	0.63
25	11830	0	0	2	0.52	4	0.64
25	11830	0	0	6	0.13	1	0.63
26	11931	0	0	2	0.55	4	0.64
26	11931	0	0	5	0.10	0	0.63
26	11931	0	0	6	0.10	0	0.63
30	42026	0	0	2	0.55	4	0.64
31	42039	0	0	6	0.10	1	0.63
32	51000	0	0	6	0.13	0	0.63
36	51211	0	0	5	0.10	1	0.63
36	51211	0	0	5	0.10	2	0.63
36	51211	0	0	5	0.13	2	0.63
37	51416	0	0	2	0.55	4	0.64
38	51428	0	0	5	0.10	1	0.63
41	51542	0	0	5	0.10	1	0.63
41	51542	0	0	5	0.10	2	0.63
41	51542	0	0	6	0.10	1	0.63

Figure 6-26: Total (100%) Recognition [Digit Two] Summary

A summary of the total (100%) recognition, i.e. chromosomes that correctly identified all the digit two contours without any false alarms, is shown in Figure 6-26.

The three main ways of observing a digit two contour for complete recognition (100%) were defined by chromosomes of the form 0-0-2, 0-0-5 and 0-0-6, with slight variations in the start search position on the line-segment vector data and the line-segment vector length threshold. Note also that the fittest chromosome, of the form 0-0-2 with fitness 0.81, did not give total (100%) recognition of the digit two contour. The test set did not produce a fitness of this value because the test set shows more variation in the digit two shapes than the training set. These results have shown that correct (100%) recognition can be achieved by a choice of chromosome, fitness threshold (Rthreshold) and the recognition measure (R).

This research work has identified a practical difficulty that occurs when starting the search of the list of input contour data. Referring to chromosome 1, Figure 6-2, the start search gene is equal to 0.19. Figure 6-27 shows the calculation for the five digit two contours shown in the second row of Figure 6-6 (Test set A) and the recognition results in the second row of Figure 6-9.

Chromosome 1			0 0 2 0.19 2		
Shape Ident & [Number of Line-Segments]	Start Search	Start Quadrant	Finish Quadrant	Line-Segment Length	Direction
Ou206 [25]	0.19x25 =4.75=5	0 2	0 0	9 45	6 7
Ou207 [24]	0.19x24 =4.56=5	0 2	0 0	5 45	6 7
Ou208 [24]	0.19x24 =4.56=5	0 2	0 0	4 35	6 7
Ou209 [36]	0.19x36 =6.84=**7**	0 0	0 0	5 3	7 6
Ou209 [36]	Truncate to **6**	0 2	0 0	**3** **39**	6 7
Ou210 [32]	0.19x32 =6.08=**6**	0 1	0 0	3 3	6 7

Figure 6-27: Start Search Calculation for Chromosome 1 (GA Test 1)

Incorrect line-segment vector lengths (5 and 3) were found for the contour ou209 when starting the search at position 7 in the input contour data. Starting the search at position 6 gave a better fitness result. See highlighted data in Figure 6-27. Incorrect line-segment vector lengths (3 and 3) were

found for the contour ou210 because the line-segment input-data is noisier than the other examples of the digit two (see ou210 in Figure 6-6). Although the start search position gave problems for ou209 and ou210, the fitness calculation provided a relatively high fitness value (second rowFigure 6-9) and thus effectively reduced the start search problem for these two input contours.

6.6 Summary and Conclusions

This research work has shown that correct (100%) recognition could be achieved, using the new technique, by a choice of chromosome, fitness threshold (Rthreshold) and the recognition measure (R). The value of R could vary for each chromosome and was thus a flexible feature of this new recognition method. The fittest chromosomes were not necessarily the best for correct recognition, and when used in combination the chromosomes could provide a recognition capability with few false alarms. A mismatch recognition capability was also provided by the genetic algorithm developed in Chapter 5, which could be used as extra evidence for contour recognition. This mismatch capability was provided without changing the genetic algorithm or the recognition process, and was achieved by a modification in the fitness calculation only. A summary of the recognition percentages versus false alarm percentages without selection of the best recognition chromosomes is shown in Figure 6-28 to Figure 6-31. These figures show the spread of the solutions without any processing of the recognition thresholds or the choice of the best recognition chromosomes. It can be seen that the false alarms can be reduced to zero by a choice of chromosome, fitness threshold (Rthreshold) and the recognition measure R.

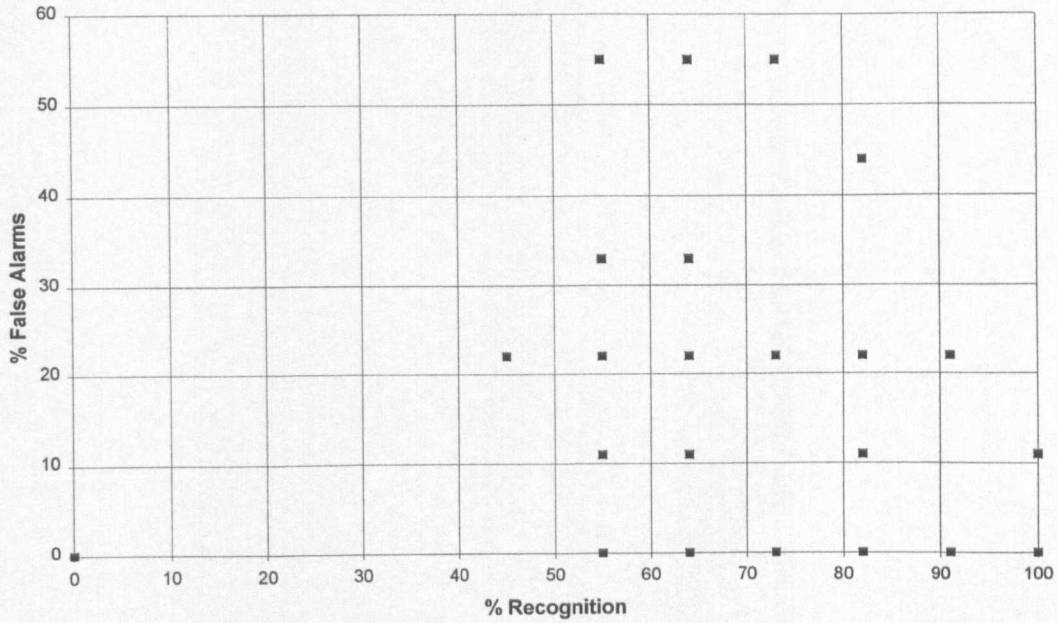

Figure 6-28: Training Set [Digit Two]: All cases: False Alarms v Recognition

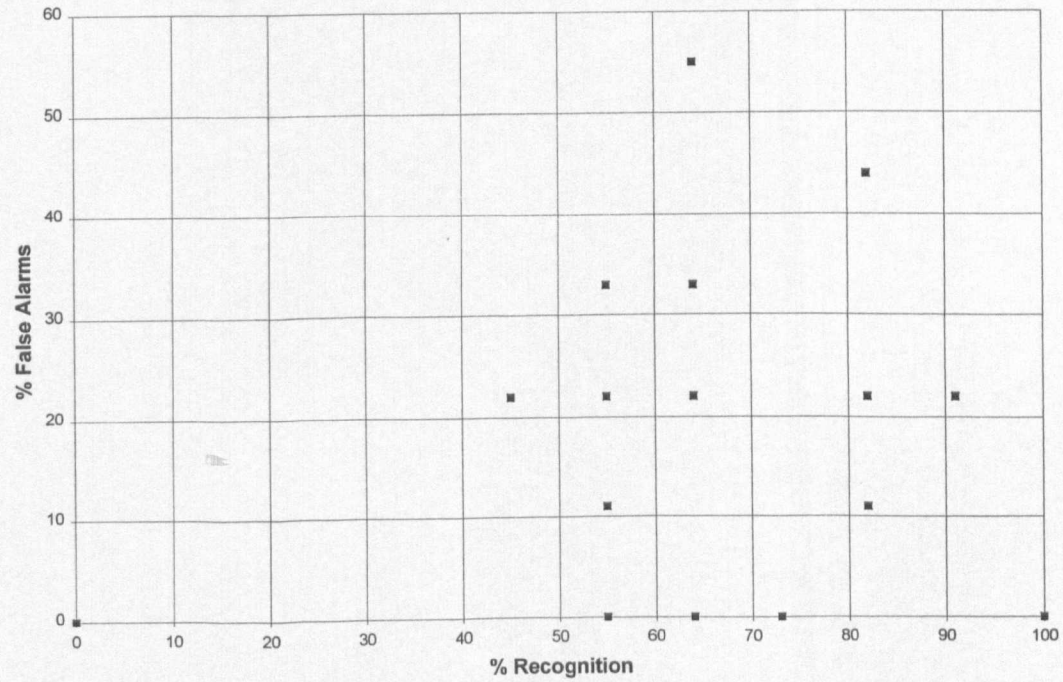

Figure 6-29: Training Set [Digit Two]: Pc = 0.6: False Alarms v Recognition

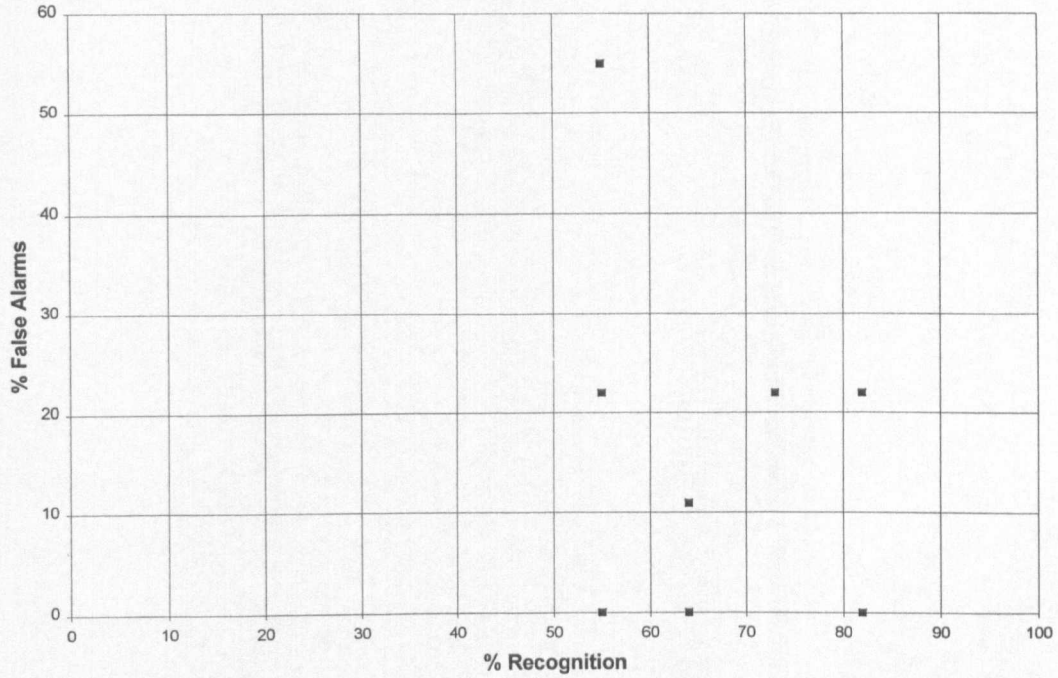

Figure 6-30: Training Set [Digit Two]: Pc = 0.0: False Alarms v Recognition

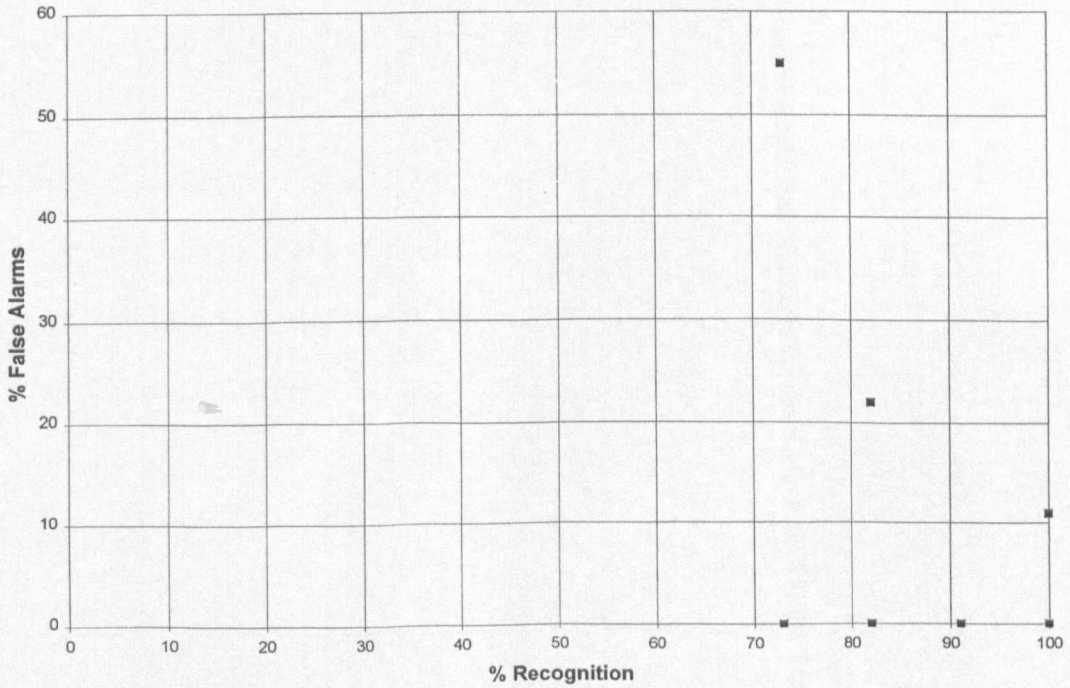

Figure 6-31: Training Set [Digit Two]: Pm = 0.0: False Alarms v Recognition

The new recognition technique developed in this chapter could be suitable for real-time applications and should be immune to contour occlusion problems. The processing involved should not be excessive and involves fitness calculations on the test contour, which are controlled by a small number of training set chromosomes. Applications could be developed where both the chromosome

and the recognition measure (R) would be evolved as the test contour is being observed. Occlusion could then be overcome because only parts of the contour, from 2 to 8 line-segments, would be examined or observed by a chromosome at any one time. The line-segment vectors along the contour, that are not occluded, may be sufficient for the recognition process to succeed. Further research is recommended to investigate contour occlusion using the new recognition technique developed in this chapter.

Certain combinations of line-segment vectors, specified by the chromosome, show high fitness but the contour shapes do not appear similar to the human observer. Likewise certain combinations of line-segments have low fitness and the contours appear to be similar. The results from the research described in this chapter have indicated that:

1. The line-segment search start point in the input contour data may be too sensitive to noise on the measured contour line-segments. Further analysis in this area of the algorithm is suggested as future research.
2. The calculation of a two dimensional fitness value can suffer from a non-linearity and/or asymmetry such that a higher fitness value can be calculated for the chromosome, but the shapes of the line-segment vectors involved are not more like those of the training set contours. A modification to the fitness function is discussed in Chapter 7, which adjusts the fitness calculated for the difference in direction of two line-segments, to take account of a clockwise or anti-clockwise difference (difference modulus (4)). This modification is shown to improve the linearity of the fitness function.

The research work reported in this chapter has shown that recognition of a contour was possible, without false alarms, using chromosomes evolved by a genetic algorithm as described in Chapter 5. Because only parts of the contour are observed, recognition of these sections of the contour would thus be less affected by occlusion than other methods.

7. Chromosome Fitness Calculation

7.1 Introduction

In Chapter 3 it was noted from the research experimental results, that there were difficulties with the fitness calculations, when reconstructing a contour and then comparing it with a reference contour. Premature convergence occurred towards a shape boundary contour that was different to the reference contour. The research work in this chapter investigated this problem and has suggested that the fitness calculations should be analysed to show that there are no local incorrect maxima in the fitness function itself. Local maxima in the fitness function will cause the genetic algorithm to bias the selection process towards the local incorrectly calculated maxima solutions at the expense of the correct solutions. The genetic algorithm is able to deal with local maxima in the environment but will have difficulties when a fitness calculation incorrectly produces an apparent local maxima.

A review of related work is presented which shows the variety in the various fitness calculations that are used by genetic algorithms. Few references discuss the suitability of the chosen fitness calculation for solving the particular problem being investigated. Matching shapes usually involves calculating an Euclidean distance error between a reference or model shape contour and the shape contour under observation. As can be seen from the references below, the design of a fitness function that correctly measures the difference between shape contours that match the 'perceptual' similarity as seen by a human observer, can be difficult.

Some of the research experiments tested examples of fitness calculations that can also generate local minima in the fitness values. These calculations biased the selection process in the genetic algorithm away from chromosomes that would contribute to the 'best' solution. This type of defect in the fitness calculation is not as serious as local (incorrect) maxima in the fitness function, but it does reduce the performance of the genetic algorithm.

The experimental test results from the various examples of fitness calculations were analysed. The

defects of the fitness calculations are discussed here. Particular attention is focused on the linearity and symmetric characteristics of the various fitness functions. Fitness calculations involving line-segment directions were observed to be dependent on the input data in some cases. Suggestions for further research on the triangular fitness function used by the genetic algorithm described in Chapters 5 are proposed.

7.2 Review of Related Work

References to various fitness functions in the literature are listed inFigure 7-1.

	Reference	Topic	Fitness Function
1	Andry and Tarroux, 1994	Image Segmentation	$f = -\sum \lvert x_i - y_i \rvert$
2	Bornholdt and Graudenz, 1992	Neural Network Structure	$f = \frac{n}{1+ka}$
3	Brunnstrom and Stoddart, 1996	Surface Matching	$Q = e^{-\left(\frac{d^2}{2\sigma^2}\right)}$ $f = e^{\left(\frac{Q}{T}\right)}$
4	Chambers, 1995	Fitness Function Design	$f(p) = \frac{1.0}{(1.0+V(p))}$
5	Chun and Yang, 1996	Region-Based Image Segmentation	$f = GE$
6	Ostrowski, 1995	Non-Linear Digital Filter Design	$f = C_{\max} - C$
7	Pelillo, 1995	Relaxation Labelling	$F = \tan\left\{0.5\pi\left(1 - E/E_{\max}\right)\right\}$
8	Saito and Mori, 1996	Object Shape Modelling	$f = E1 - \alpha E2$
9	Singh, 1997	Structural Shape Matching	$f = (K - Cost)^n$ $f = \left(\frac{K}{Cost}\right)^n$
10	Toet and Hajema, 1995	Contour Matching	$f(M,E) = \sqrt{\frac{1}{n}\sum_1^n \left(v_M - v_E\right)^2}$
11	Yip and Pao, 1994	Quadratic Assignment	$\text{cost} = \sum ab$

Figure 7-1: Fitness Calculation References

The form of typical fitness calculations is presented in the right hand column. The calculations compare a measured with a reference or model variable, such that the difference is an indicator of

the fitness. Fitness function 10 is popular in the literature, and gives an *rms* error that reduces to zero as the measured variables match the model variables. The 'best' solutions will in this case have minimum fitness which approaches zero as the variables coincide.

A selection of references is presented below which identify various potential problems associated with the fitness calculation and give some guidelines on how a fitness function should be designed. The linearity and symmetry properties of the fitness calculations appear to be of major importance to the correct operation of the fitness function. In the context of shape recognition, linearity means that as the fitness increases the similarity between shapes follows the observations of a human observer. Symmetry refers to the ability of the fitness function to enable chromosomes to be ranked (ordered) satisfactorily for the genetic algorithm selection process. The chromosomes will then be chosen correctly for reproduction in proportion to their true relative fitness values.

Kupeev and Wolfson (1996) described a measure of 'perceptual' similarity among shapes which 'look' similar. Shapes are presented as special weighted graphs, the vertices of which represent the 'lumps' of the shapes in a given orientation. These graphs are then reduced, using a trimming procedure, until the resulting graphs are isomorphic. This paper notes that some of the distance functions, often Euclidean, used by the various recognition methods ensure that if the distance between the representations is approximately equal to zero, then the corresponding contours are approximately equal. "Nevertheless, these characteristics (distance functions) do not automatically ensure that lower values of the distance calculated testify to the greater perceptual similarity of the represented shapes."

Chun and Yang (1996) presented a region-based image segmentation methodology using a genetic algorithm. A fuzzy validity function is proposed, which measures a degree of separation and compactness between and within finely segmented regions and an edge-strength along boundaries of all regions. This paper notes that this type of fitness measure is not guaranteed to be optimal as a

universal measure of segmentation quality. In the case of a complicated image, which includes unclear boundaries among regions and texture characteristics, the fitness is continuously fluctuating, because it cannot find an optimal solution.

Ostrowski (1995) addressed the estimation approach to non-linear digital filter design using a genetic algorithm. The problem can be stated as selecting a filter from a definite class of window width such that the cost function under some assumed error criterion, e.g. the mean absolute error or the mean squared error between the output of the filter and the desired signal, is minimised. This paper also discusses the use of a non-linear scaling function.

Toet and Hajema (1995) formulated object recognition as an optimisation problem, where the objective function measures the evidential support for any particular projection of a parameterised object contour model onto an input image contour. No single universal evaluation function exists for all matching problems. The objective function used by Toet and Hajema (1995) was based on the observation that a good match is one where every element of the projected model contour is spatially near an image contour. The objective function is formulated as "the cumulative sum of the Euclidean distance values along the projection of the model contour in the image plane." The root mean square average is stated as a good choice for the distance measure between an edge image and the projected model contours. Other possible Euclidean distance measures are noted, such as the median of the distance values for the individual contour points, or the fraction of the number of model edge points that are within a certain threshold distance from the edges in an image.

Yin (1998) proposed a new polygonal approximation method that used a genetic algorithm to determine the optimal polygons that can be fitted to a digital contour. This method used a fitness calculation similar to fitness function 8 in Figure 7-1, where E2 was either the maximum error or the integral square error in the perpendicular distance between the individual polygon line-segments and the actual contour points. Fitness scaling was used to reduce the possibility of premature

convergence and the value of E1 was derived from the maximum and minimum values of E2 in each generation. Good approximations were obtained, compared to other algorithms, but no discussion was provided as to which errors to use or whether the variation in E1 improved the selection process.

D. Corne in Chambers (1995, page 230) noted that "... except for artificial test problems that are thoroughly understood, it is difficult or impossible to design a fitness function whose graph will be best explored by a given set of genetic operators. After all, if the problem is that well understood, there would be no need to use a genetic algorithm to solve it in the first place. However, there are some natural choices available for the fitness function. In a timetable problem one might choose to use something inversely proportional to the number of violated constraints (see fitness function 4, Figure 7-1). However, this function treats all constraints equally and conveys no information about the extent to which any is violated. A better choice can be made."

Kinnear (1994, page 9, section 1.4.2) discussed the *exceptional* importance of the fitness function. "You simply cannot take too much care in crafting your fitness function. Any and all boundary conditions, in your fitness function, will be exploited ruthlessly by the individuals in your population. Outright though subtle bugs in your fitness calculations will almost certainly be recognised by the evolutionary process, and the only way in which this can be determined is to examine the individuals that evolve." Kinnear (1994), in the same section of the above reference, also highlighted the following characteristics required by a fitness calculation:

1. A central aspect of fitness functions is the concept of *partial credit.* A fitness function needs to score an individual on how well it performs on the problem being solved. "However it needs to do more than this, it needs to score individuals in such a way that they can be compared and a more successful individual can be distinguished from a less successful individual."
2. Any fitness function must be able to evaluate partial solutions to a problem and produce a score that distinguishes a more complete solution from a less complete solution. The favouring of one

partial solution over another defines the direction that the evolutionary processes encourage individuals in the population to move.

3. In addition to partial credit, there are other factors to consider when designing a fitness function. "One of these factors is how to aggregate the results of multiple fitness tests, in those cases where the overall fitness of an individual is determined by its performance during several separate fitness tests."

This last point is relevant to the design of the fitness function in the research described in Chapter 5. The total measured fitness is calculated from the sum of the average fitness of the individual line-segment features, and the finish quadrant fitness values.

The review of related research work described above has shown that a variety of fitness calculations have been proposed in the literature. Very few of these referenced papers discussed the suitability of the fitness function for the solution to the relevant problem, and often assumed that an *rms* Euclidean difference calculation is suitable for the problem at hand.

Parizeau (1998) described a genetic algorithm method for optimising the cost-matrix of any approximate string-matching algorithm and *does* discuss the problems associated with various objective or fitness functions. Optimising techniques like gradient descent cannot be used for the problem discussed in this paper, because the function to be optimised, which depends on an editing distance, does not have an analytic form and thus cannot be differentiated. This paper observed that many different cost matrices could have almost identical discrimination power especially if the classes are non-homogeneous. This is one of the few papers that thoroughly discussed the form of the objective or fitness function.

The triangular fitness function used by the genetic algorithm in Chapter 5 is analysed below (see also Section 7.3). A modification to the fitness function is investigated which added a refinement to

the fitness calculation of the line-segment direction fitness. Figure 7-2 shows the line-segment directions for a reference shape (0) and two candidates for matching (3 and 5). A fitness calculation using a clockwise rotation calculates the values for the direction differences Δ_1 *and* Δ_2 to be 3 and 5 respectively, as shown in Figure 7-3.

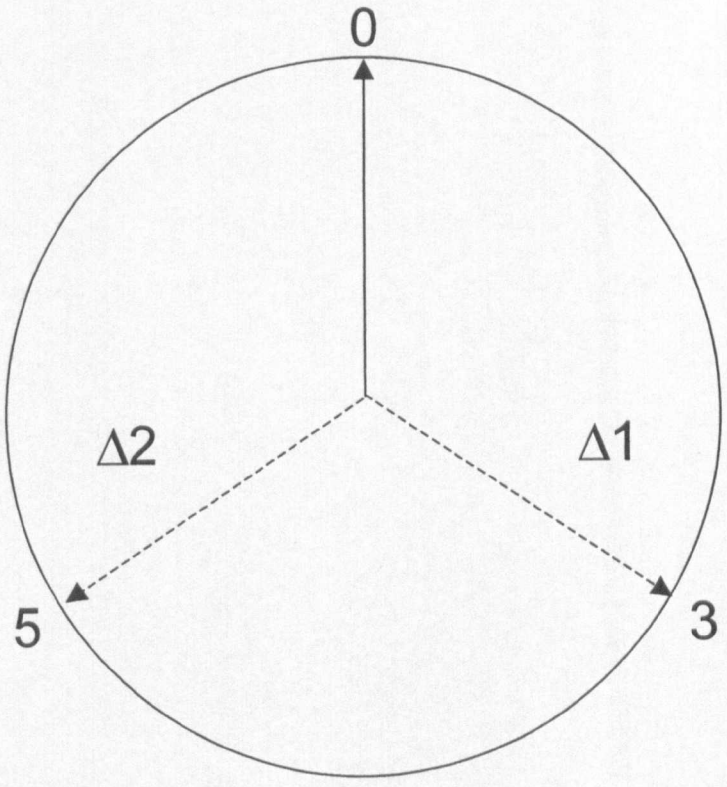

Figure 7-2: Line-Segment Directions

$$\Delta_1 = |0-3| = 3$$
$$\Delta_2 = |0-5| = 5$$
$$\textit{if } \Delta > 4 \textit{ then } \Delta = 8-|\Delta|$$
$$\therefore\ \Delta_2 = 8-5 = 3$$

Figure 7-3: Modulus (4) Modification

This fitness calculation biases the selection process in favour of the line-segment with direction 3, rather than giving equal opportunity to the line-segment with direction 5. A modification to the fitness function is shown in Figure 7-3. This modification checks for differences greater than 4 and

adjusts the direction differences Δ_1 *and* Δ_2 to each be equal to 3. This fitness function now gives equal selection opportunity to each of the line-segments shown in Figure 7-2. This modification is referred to as the 'modulus (4)' modification, in the text below, and adjusts the fitness function to use a clockwise and an anti-clockwise rotation when calculating the fitness of a line-segment direction.

7.3 Fitness Calculation Tests

The research work discussed in this chapter investigated various fitness functions (listed below) to show that it is important and necessary that the variation of the fitness with the appropriate parameter(s) is analysed. The tests showed that certain difficulties could occur, especially when the fitness calculation involved two dimensional shape boundary contours. The linearity and symmetry of the fitness function and its possible effect on the selection process in a genetic algorithm are also discussed.

The research work performed the following experiments (tests) in order to analyse various fitness functions used in the research described in Chapters 3, 5 and 6:

1. A fitness calculation for a low-pass filter step-response (Figure 7-4).
2. A fitness calculation for shape 1 using moments and shape centre of gravity (Figure 7-5). All the test shapes have the same length of perimeter as the reference shape.
3. A fitness calculation for shape 2 using moments and shape centre of gravity (Figure 7-6). All the test shapes have a longer perimeter than the reference shape.
4. A fitness calculation for shape 3 using moments and the shape centre of gravity (Figure 7-7). All the test shapes have the same length line-segments but the second line-segment has varying directions.
5. A fitness calculation for shape 2 using the genetic algorithm triangular fitness function, without and with the modulus (4) modification (Figure 7-8).

6. A fitness calculation for shape 3 using the genetic algorithm triangular fitness function, without and with the modulus (4) modification (Figure 7-9).
7. A fitness calculation for shape 4 using the genetic algorithm triangular fitness function, without and with the modulus (4) modification (Figure 7-10).
8. A test of the behaviour of the genetic algorithm developed in Chapter 5 using the modulus (4) modified triangular fitness function. The results are shown inFigure 7-18 to Figure 7-21.

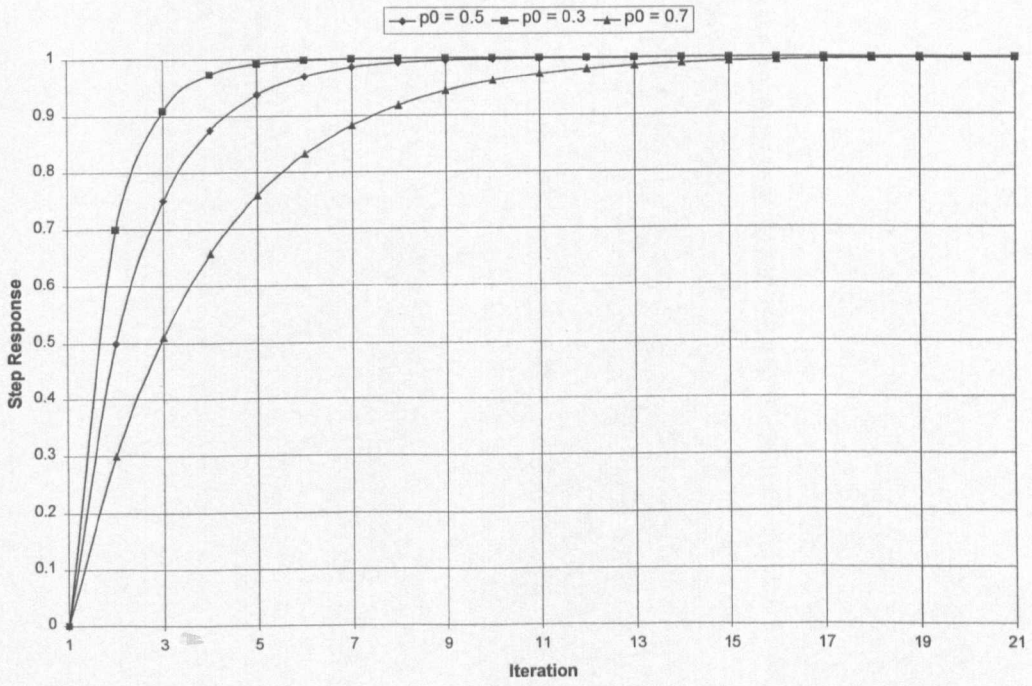

Figure 7-4: Low Pass Filter Step-Response for Various Values of P0

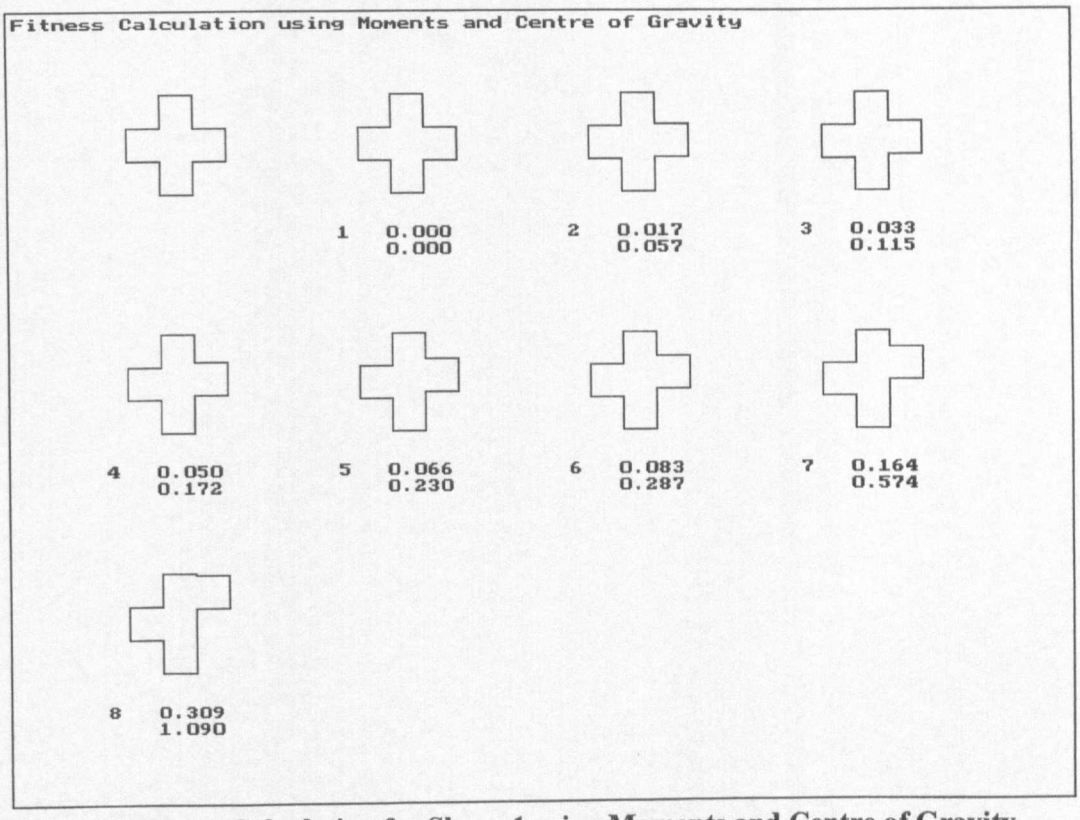

Figure 7-5: Fitness Calculation for Shape 1 using Moments and Centre of Gravity

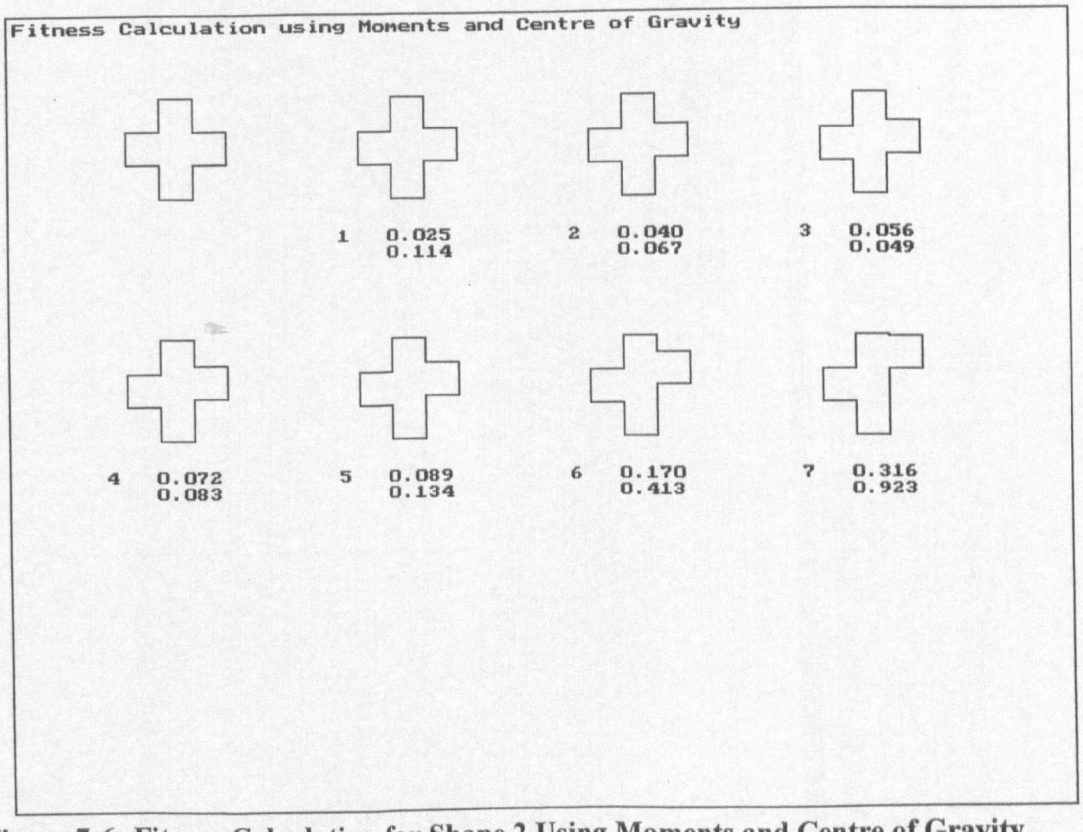

Figure 7-6: Fitness Calculation for Shape 2 Using Moments and Centre of Gravity

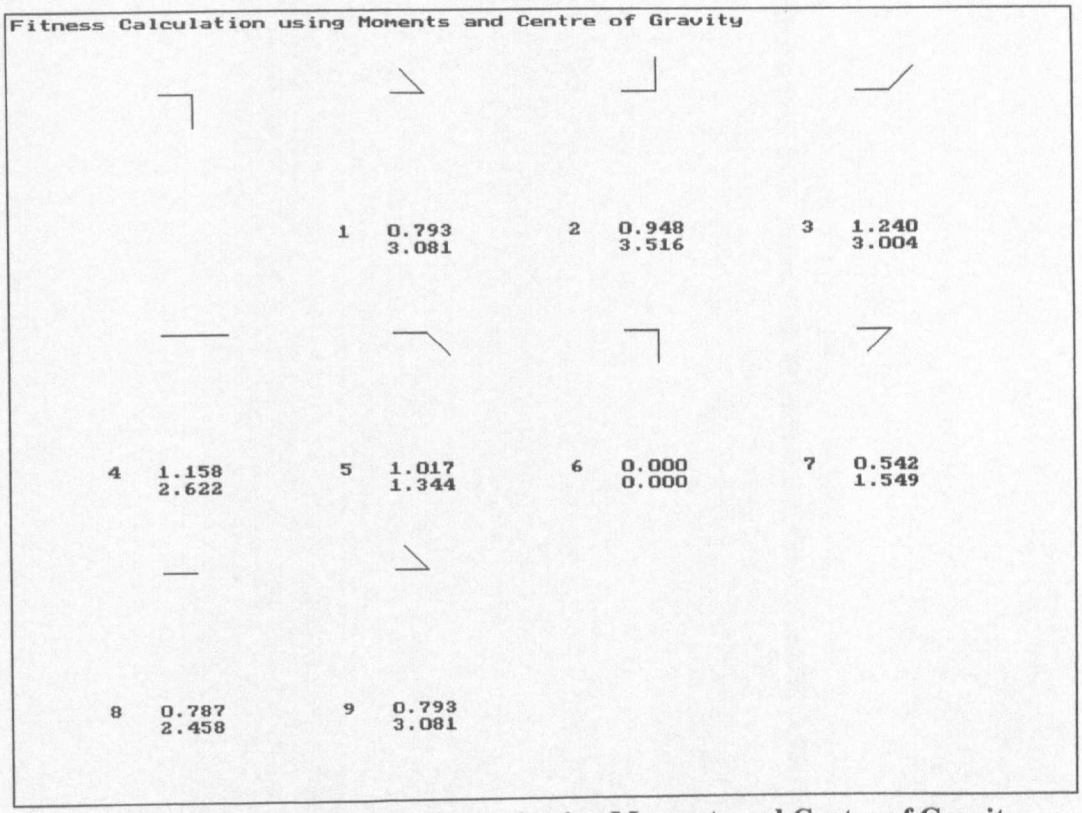

Figure 7-7: Fitness Calculation for Shape 3 using Moments and Centre of Gravity

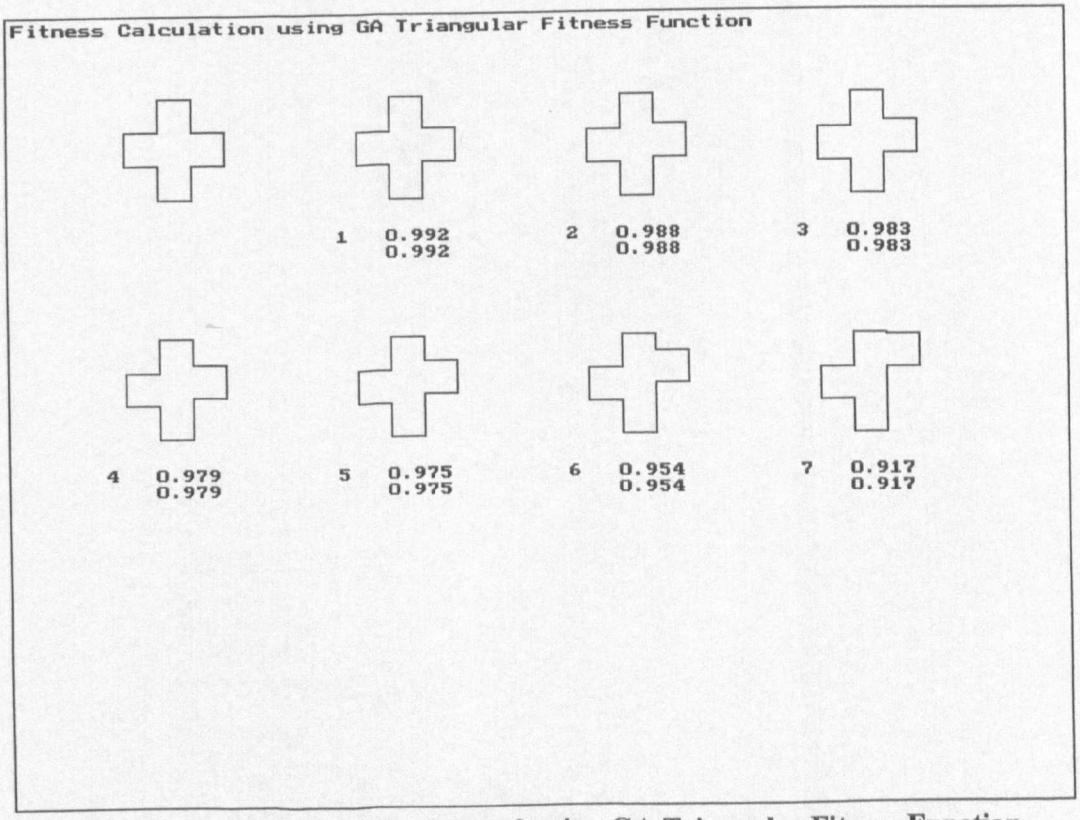

Figure 7-8: Fitness Calculation for Shape 2 using GA Triangular Fitness Function

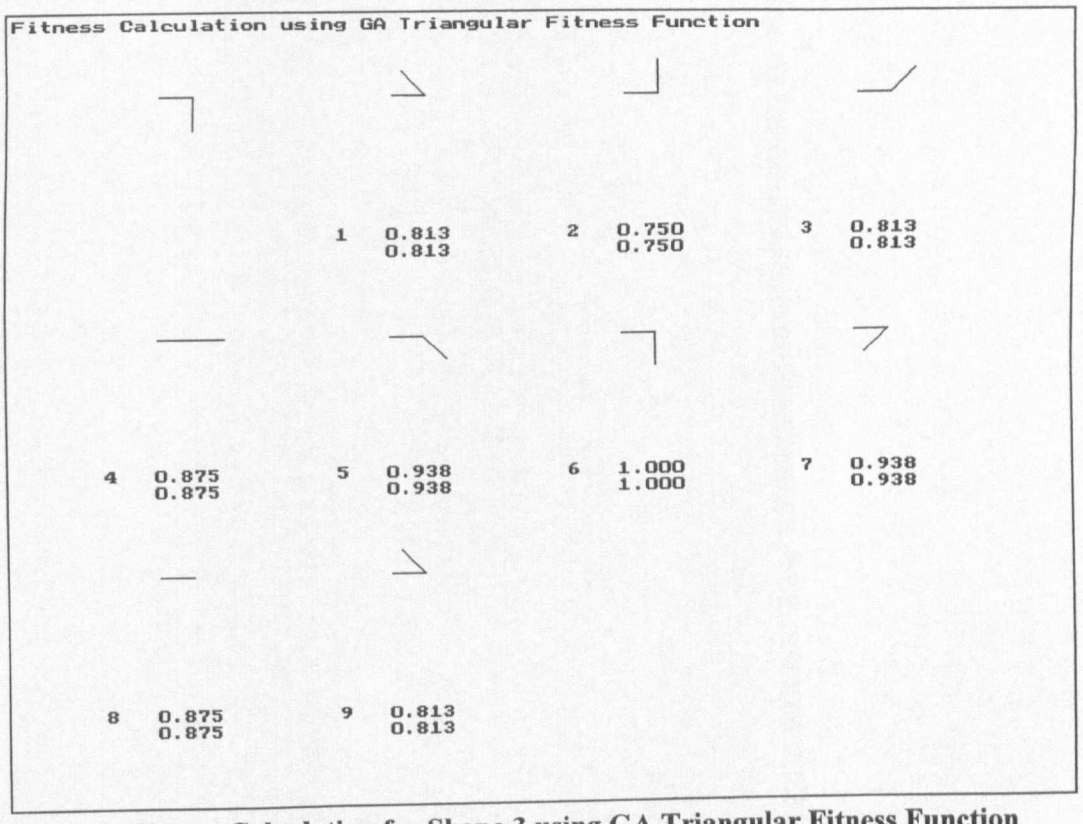

Figure 7-9: Fitness Calculation for Shape 3 using GA Triangular Fitness Function

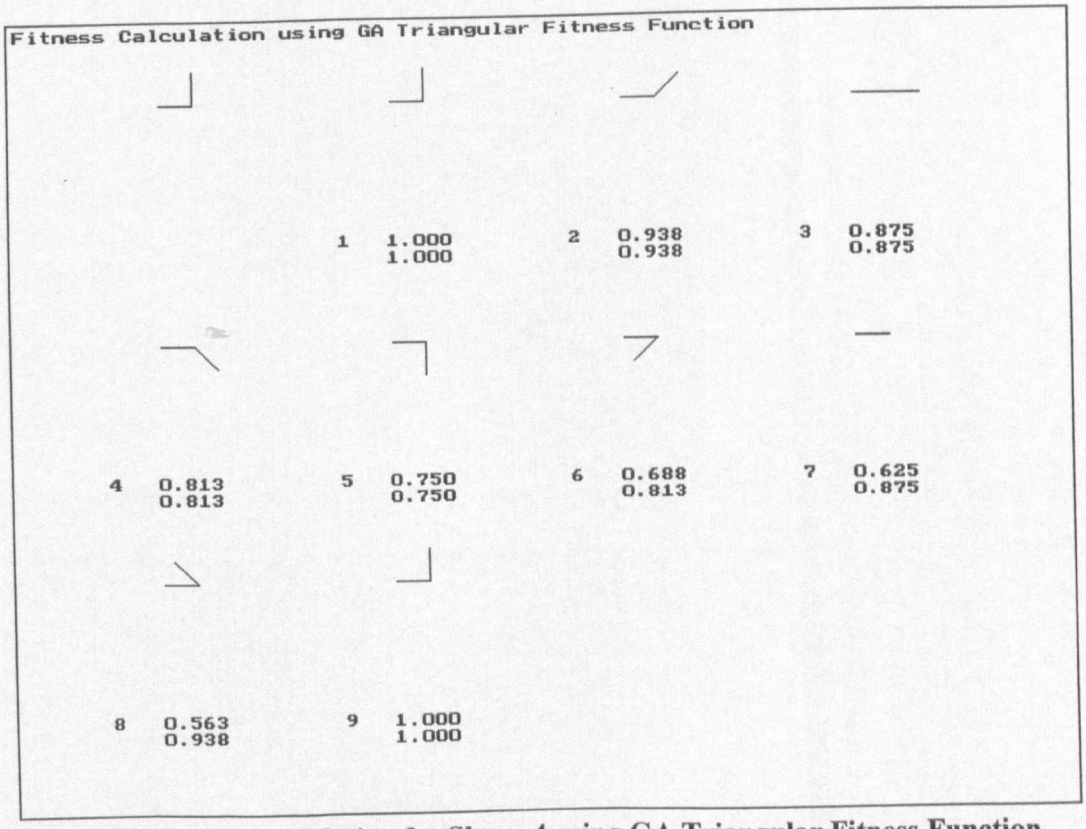

Figure 7-10: Fitness Calculation for Shape 4 using GA Triangular Fitness Function

7.4 Discussion of Test Results

7.4.1 Low Pass Filter Test

A digital low-pass filter can be implemented by the following equations:

$$y(n) = p_0 y(n-1) + p_1 x(n) \tag{7.1}$$

$$p_1 = 1 - p_0 \tag{7.2}$$

where y is the filter output, x is the filter input and n is the current iteration.

The reference step-response of the low-pass filter is calculated for $p_0 = 0.5$, and is shown in Figure 7-4. The fitness for the step-response of the low-pass filter is calculated for values of p_0 between 0.0 and 1.0. The fitness of the step-response, for a particular value of p_0, is calculated as the *rms* error between the reference step-response and the step-response of the filter for that particular value of p_0. The error is calculated as the Euclidean distance between the reference step-response and the output of the filter for the particular p_0 value for a fixed length of the filter output. The variation of step-response fitness for the range of p_0 values is shown in Figure 7-11. The step-response fitness can be seen to be asymmetrical. Chromosomes representing $p_0 > 0.7$ are heavily penalised in the selection process. The variation of fitness for values of p_0 between 0.0 and 0.5 is only 0.05, therefore the chromosomes in this range lack sufficient differentiation for the selection process. A ranking of the chromosome fitness should be applied to this problem, with possibly a different slope either side of the p_0 parameter, which gives the maximum fitness in any one generation.

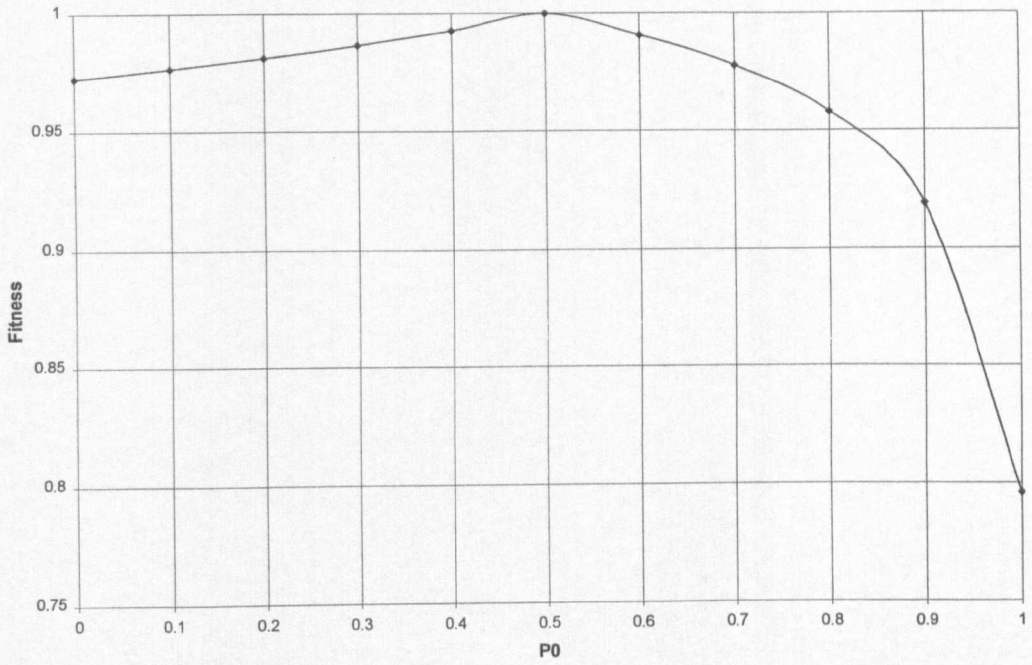

Figure 7-11: Low Pass Filter Step-Response Fitness Variation

7.4.2 Shape 1 Moment Descriptor Test

Figure 7-5 shows the fitness calculations using moment descriptors and the centre of gravity (centroid) for Shape 1. The upper numbers are the *rms* error between each of the normalised central moments for each test shape, and the reference shape in the top left-hand corner of the display. The lower numbers include the difference between the x and y centre of gravity values for each shape and the reference shape. The fitness variation for the different test shapes is shown in Figure 7-12, where an *rms* error of 0.0 is normalised to a fitness of 1.0. The variation can be seen to be non-linear with essentially two different slopes that change after test shape 6. Test shapes 6 to 8 are some 'distance' from the reference shape, which is shown by a sharp reduction in the fitness value for these test shapes. The variation in fitness for test shapes 1 to 5 is linear but ranking could be applied to provide a wider range for the selection process. The variation of fitness with test shape for this case, although non-linear, will be satisfactory, because the non-linearity works in favour of the correct solution, i.e. a linear region for solutions near to the reference shape and a sharp, but linear, decrease for solutions far from the reference shape.

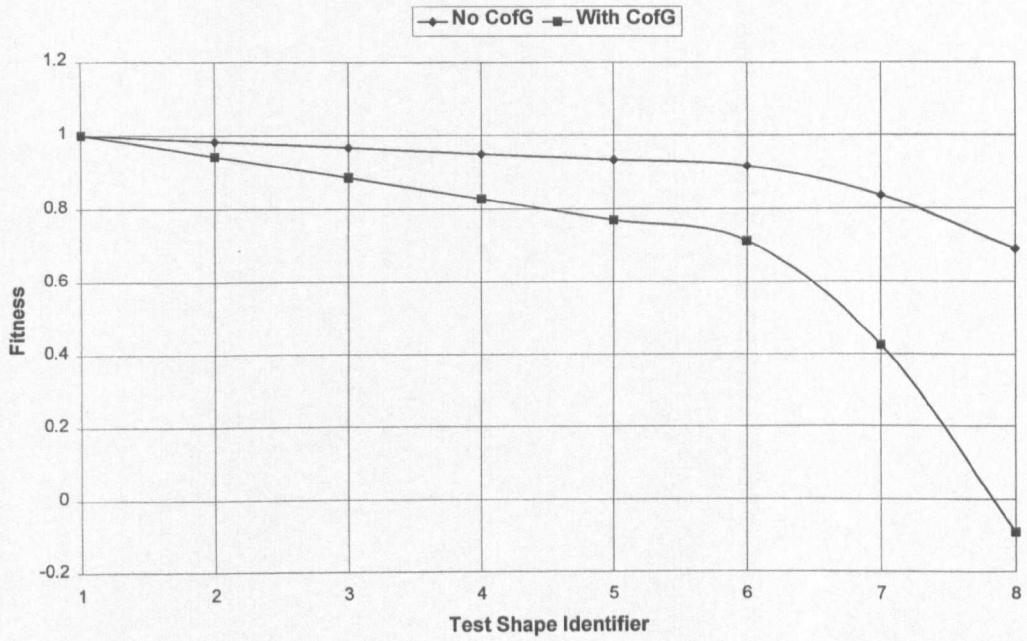

Figure 7-12: Fitness Variation for Shape 1

7.4.3 Shape 2 Moment Descriptor Test

Figure 7-6 shows the fitness calculations using moment descriptors and the centre of gravity (centroid) for Shape 2 and is in the same format as Figure 7-4. Shape 2 test shapes have a perimeter longer than the reference shape. The fitness variation for the different test shapes is shown in Figure 7-13, where an *rms* error of 0.0 is normalised to a fitness of 1.0.

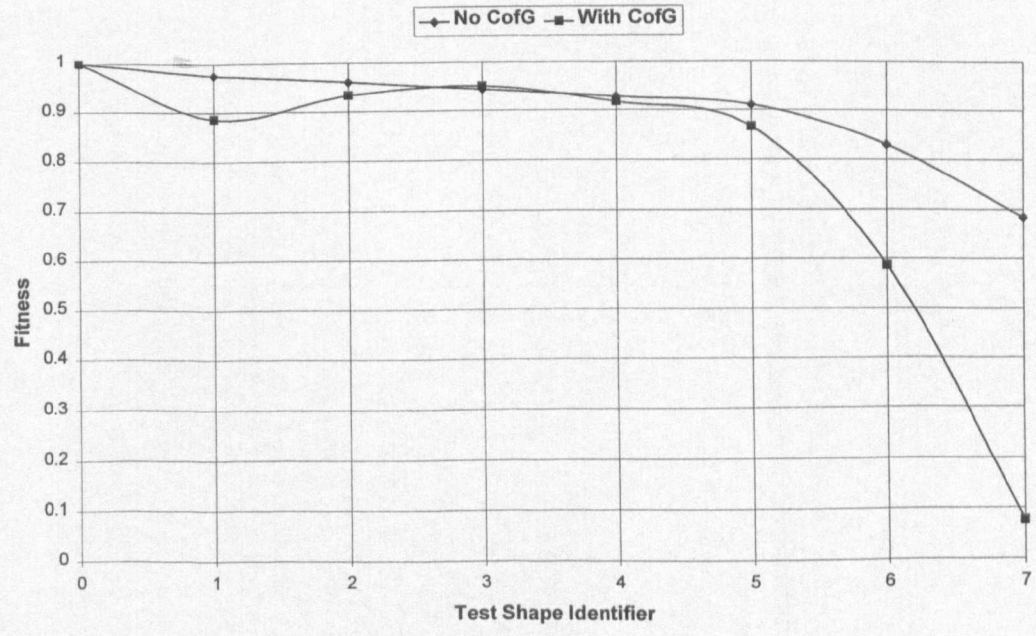

Figure 7-13: Fitness Variation for Shape 2

The variation using the moment calculations alone shows the same form of non-linearity as for Shape 1 (Figure 7-12), but the variation including the centre of gravity difference shows a local maximum (local minimum in the *rms* error) at test shape 3. This effect was also noticed in the research work in Chapter 3, where the genetic algorithm developed solutions that became locked onto a shape that was slightly different to the reference shape. Analysis of this fitness function, where the centre of gravity error is included, shows that it is not suitable for providing solutions that are the same as the reference shape. An advantage of the genetic algorithm is that local maxima in the environment have less effect on its ability to find the global maximum than other algorithms, e.g. hill-climbing techniques. In this test case an analysis of the fitness calculation shows that there is a local maxima in the fitness function that can be removed with the correct choice of fitness calculation. The performance of the genetic algorithm will then be improved.

7.4.4 Shape 3 Moment Descriptor Test

Figure 7-7 shows the fitness calculations using moment descriptors and the centre of gravity (centroid) for Shape 3 and is in the same format as Figure 7-5. The fitness variation for the different test shapes is shown in Figure 7-14, where the *rms* error of 0.0 is normalised to a fitness of 1.0.

In this case, both methods for calculating the *rms* error give a non-linearity and asymmetry about the maximum fitness shape (test shape 6). Note also that using moments alone gives a minimum in the fitness at test shape 3 and a minimum in the fitness at test shape 2 when the centre of gravity error is included in the fitness calculation. A minimum in the fitness function, for a particular shape, will mean that a bias will be applied to that shape in the selection process and these shapes will be inhibited from reproducing. The fitness of test shape 1 is greater than that for test shape 5 when using the moments only to calculate the fitness of a test shape. Hence more examples of test shape 1 could well be developed by a genetic algorithm in preference to test shape 5 which is, in fact, nearer to the reference shape in appearance, i.e. only one direction unit away from the solution. Analysis of this fitness function when used for shapes that are not closed and have few line-segments shows that

it will exhibit a bias against certain shapes. Other shapes that are further away from the solution will be enhanced. This type of fitness function exhibits local minima in the fitness calculation and will bias the genetic algorithm's search away from shapes which are more like the reference shape as seen by a human observer.

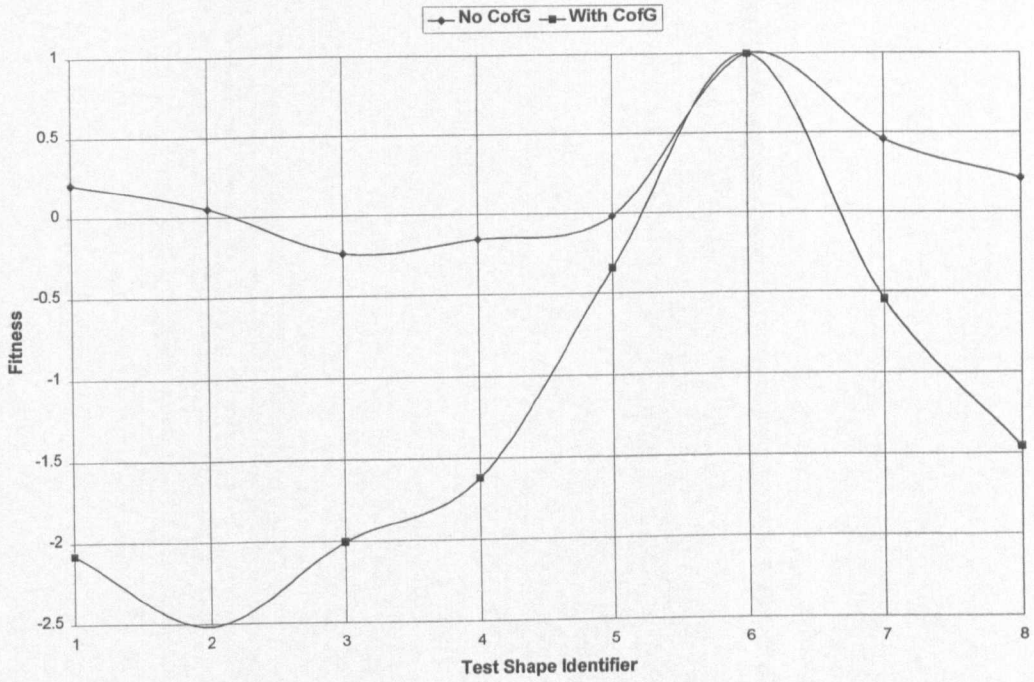

Figure 7-14: Fitness variation for Shape 3

7.4.5 Shape 2 Triangular Fitness Test

Figure 7-8 shows the fitness calculation for Shape 2 using the genetic algorithm triangular fitness function. The reference shape is in the top left-hand corner of the display. The upper set of values are calculated using the same triangular fitness function as that used by the genetic algorithm in Chapter 5 and the recognition process in Chapter 6. The lower set of values is calculated using the triangular function with the modulus (4) modification when calculating the contribution to the fitness from the line-segment directions. The fitness variation for the different test shapes is shown in Figure 7-15.

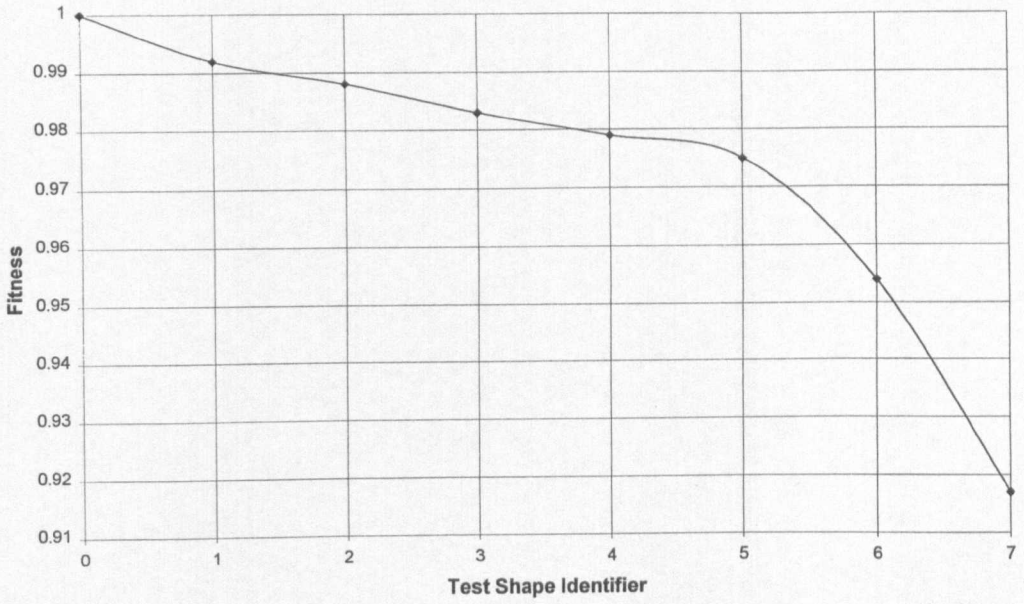

Figure 7-15: Fitness Variation for Shape 2

The fitness calculation is the same for the triangular fitness function without and with the modulus (4) modification. This fitness function exhibits the same non-linear characteristics as shown in Figure 7-12 and in Figure 7-13, for shape 1 and shape 2. The same conclusions can, therefore, be drawn. Both versions of the triangular fitness function can be used for shapes of this type.

7.4.6 Shape 3 Triangular Fitness Test

Figure 7-9 shows the fitness calculation for Shape 3 using the genetic algorithm triangular fitness function. The reference shape is in the top left-hand corner of the display. The upper set of values are calculated using the same triangular fitness function as that used by the genetic algorithm in Chapter 5 and the recognition process in Chapter 6. The lower set of values is calculated using the triangular function with the modulus (4) modification when calculating the contribution to the fitness from the line-segment directions. The fitness variation for the different test shapes is shown in Figure 7-16.

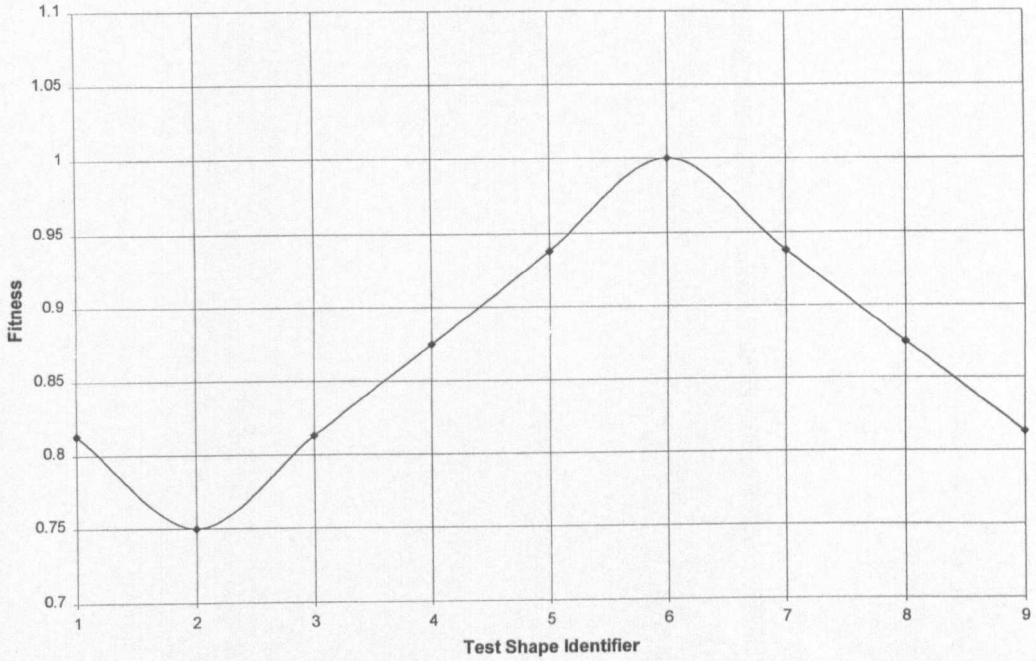

Figure 7-16: Fitness Variation for Shape 3

The fitness calculation is the same for the triangular fitness function without and with the modulus (4) modification. The fitness variation appears to be linear for both variations of the fitness calculations. The fitness for test shapes 1 and 3 are the same because the second line-segment, on the reference shape, has a direction value of 4 (South). The difference between the line-segment directions for test shapes 1 and 6 is 3 and the difference for test shapes 3 and 6 is also 3. The fitness function without and with the modulus (4) modification, in these example test shapes, gives the same fitness values. A possible fault in the fitness function is not noticed. The worst fitness is calculated for the test shape 2, which is consistent with the view of a human observer, because the second line-segment for test shape 2 is in the opposite direction to that of the reference shape.

7.4.7 Shape 4 Triangular Fitness Test

Figure 7-10 shows the fitness calculation for Shape 4 with the fitness variation displayed in Figure 7-17.

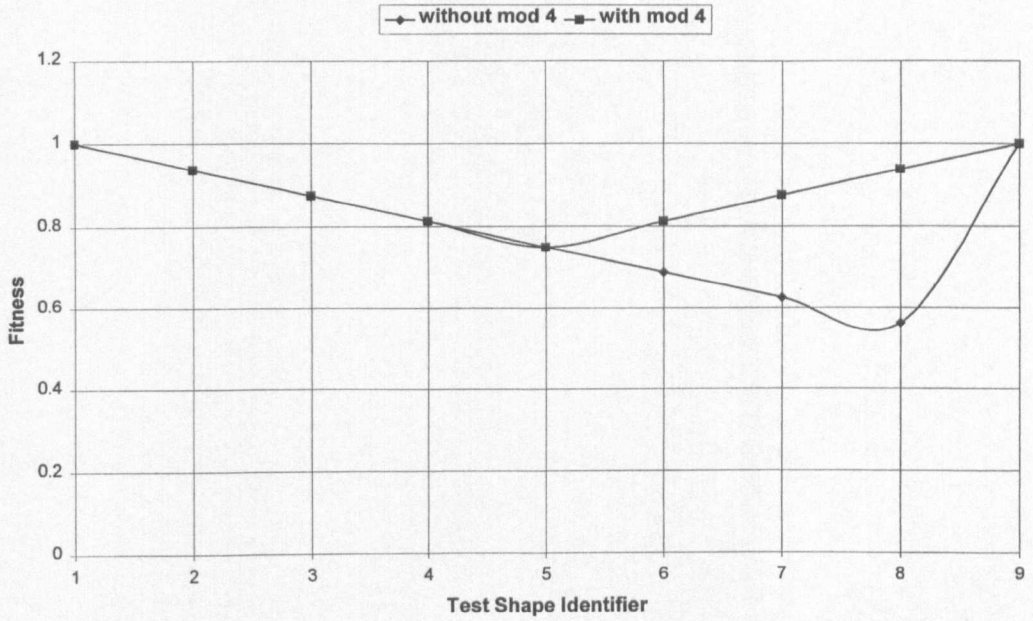

Figure 7-17: Fitness Variation for Shape 4

The fitness variation without the modulus (4) variation exhibits a reduction in the fitness as the second line-segment rotates from the 0 (North) direction to the 7 (north-west) direction (Figure 7-2). The reference direction for the second line-segment is 0 (North) and, therefore, as the second line-segment for the test shapes increases the direction value, the difference also increases. The fitness calculation using the modulus (4) variation reduces the difference between the directions of the second line-segment, as the difference passes through the value 4 (South). The test shapes 6, 7 and 8 have reduced fitness values (without the modulus (4) modification) and, thus, have a lower probability of being chosen by the selection process in a genetic algorithm. A local minima has appeared in the output of the fitness function for these particular test shapes and, without the modulus (4) modification, the fitness function is dependent on the values of input data. Test shape 5 gives the lowest fitness for both versions of the fitness function which is consistent with the view of a human observer, because the second line-segment for test shape 5 is in the opposite direction to that of the reference shape.

7.4.8 Genetic Algorithm Behaviour with Modified Fitness Function

The genetic algorithm developed in the research described in Chapter 5 was adapted to use the

modulus (4) modification to the triangular fitness function.Figure 7-18 shows the twenty fittest chromosomes produced on a particular run of the modified genetic algorithm, using the digit two training set (Chapter 5) and test sets A and B (Chapter 6).

Number	Chromosome					Fitness
1	0	0	2	0.10	0	0.81
2	0	0	2	0.10	1	0.81
3	0	0	2	0.13	2	0.81
4	**0**	**0**	**6**	**0.13**	**0**	**0.63**
5	**0**	**0**	**6**	**0.13**	**1**	**0.63**
6	3	1	2	0.39	0	0.61
7	3	1	3	0.39	1	0.60
8	0	0	2	0.61	0	0.59
9	0	1	6	0.13	1	0.59
10	0	0	2	0.06	0	0.58
11	0	1	6	0.13	2	0.57
12	3	1	2	0.39	3	0.57
13	0	0	2	0.45	1	0.54
14	1	0	2	0.13	0	0.50
15	0	0	4	0.10	0	0.49
16	3	1	2	0.32	10	0.49
17	3	1	2	0.26	10	0.49
18	3	1	2	0.32	9	0.49
19	3	0	2	0.55	3	0.48
20	0	0	6	0.19	2	0.45

Figure 7-18: Chromosome List

Seven chromosomes with different observation genes, i.e. ways of observing the contour line-segments, were evolved with chromosomes 4 and 5 providing 100% recognition (Figure 7-19).

A value of the recognition measure threshold R in the recognition matrix (Figure 7-19 and Chapter 6), with a value in the range 0.0 to 0.30, would also give 100% recognition for all twenty chromosomes. The various line-segment vectors involved in the recognition process for chromosomes 4 and 5 are shown in Figure 7-20 and Figure 7-21. The use of the modulus (4) modification to the triangular fitness function appears to have improved the performance of the genetic algorithm (see Chapter 5 and Chapter 6) for this particular training and test sets.

	Ident	Chromosome																				g	b	R
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20			
1	Two06	1	1	1	**1**	**1**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20	0	1.00
2	Two07	1	1	1	**1**	**1**	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	19	1	0.90
3	Two08	1	1	1	**1**	**1**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20	0	1.00
4	Two09	1	1	1	**1**	**1**	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	18	2	0.80
5	Two10	1	1	1	**1**	**1**	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	19	1	0.90
6	Ou206	1	1	1	**1**	**1**	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	17	3	0.70
7	Ou207	1	1	1	**1**	**1**	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	13	7	0.30
8	Ou208	1	1	1	**1**	**1**	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	15	5	0.50
9	Ou209	1	1	1	**1**	**1**	0	0	1	0	1	0	1	1	0	1	1	1	1	1	0	14	6	0.40
10	Ou210	1	1	1	**1**	**1**	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	17	3	0.70
11	A10-1	1	1	1	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	9	11	-0.10
12	A10-2	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	4	16	-0.60
13	A10-3	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	3	17	-0.30
14	A10-4	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	1	0	5	15	-0.50
15	A10-5	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	5	15	-0.50
16	A10-6	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	4	16	-0.60
17	Thomas	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	0	0	7	13	-0.30
18	Ou201a	1	1	1	**1**	**1**	1	1	1	0	1	0	1	1	1	0	0	0	1	1	1	15	5	0.50
19	Tank01	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	0	5	15	-0.50
20	Lake01	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1	0	6	14	-0.40

Figure 7-19: Recognition Matrix with Modified Fitness

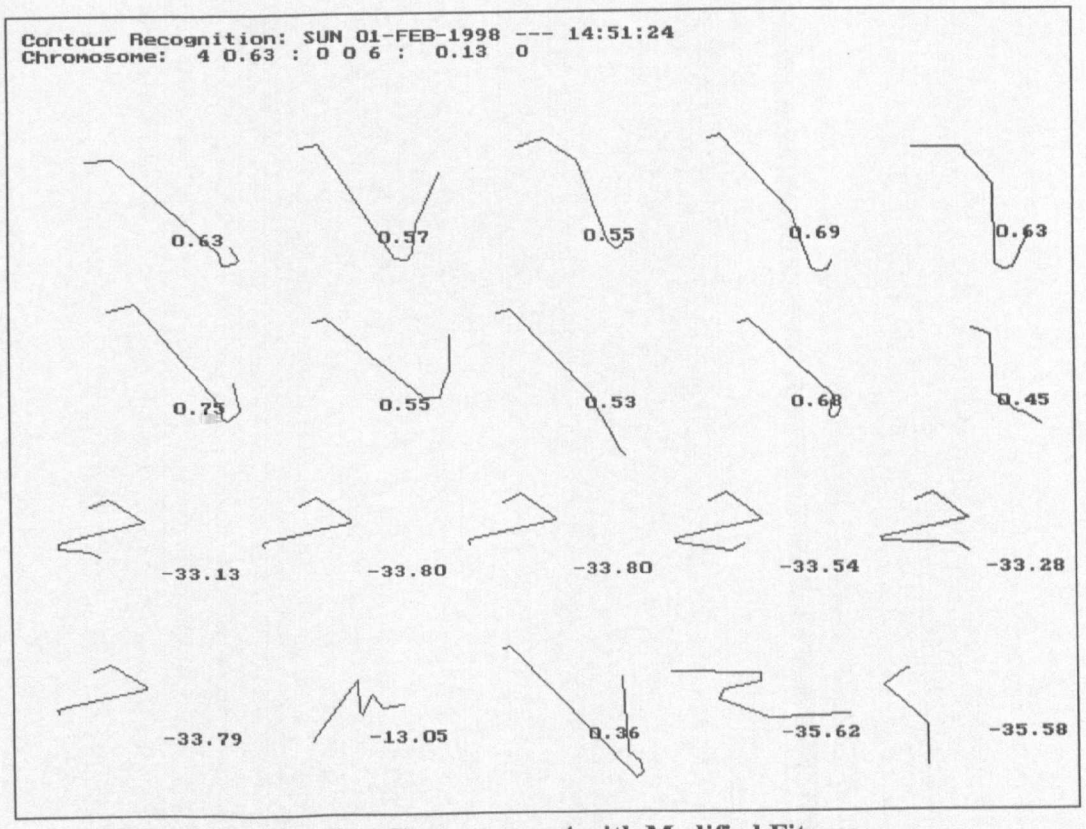

Figure 7-20: Recognition using Chromosome 4 with Modified Fitness

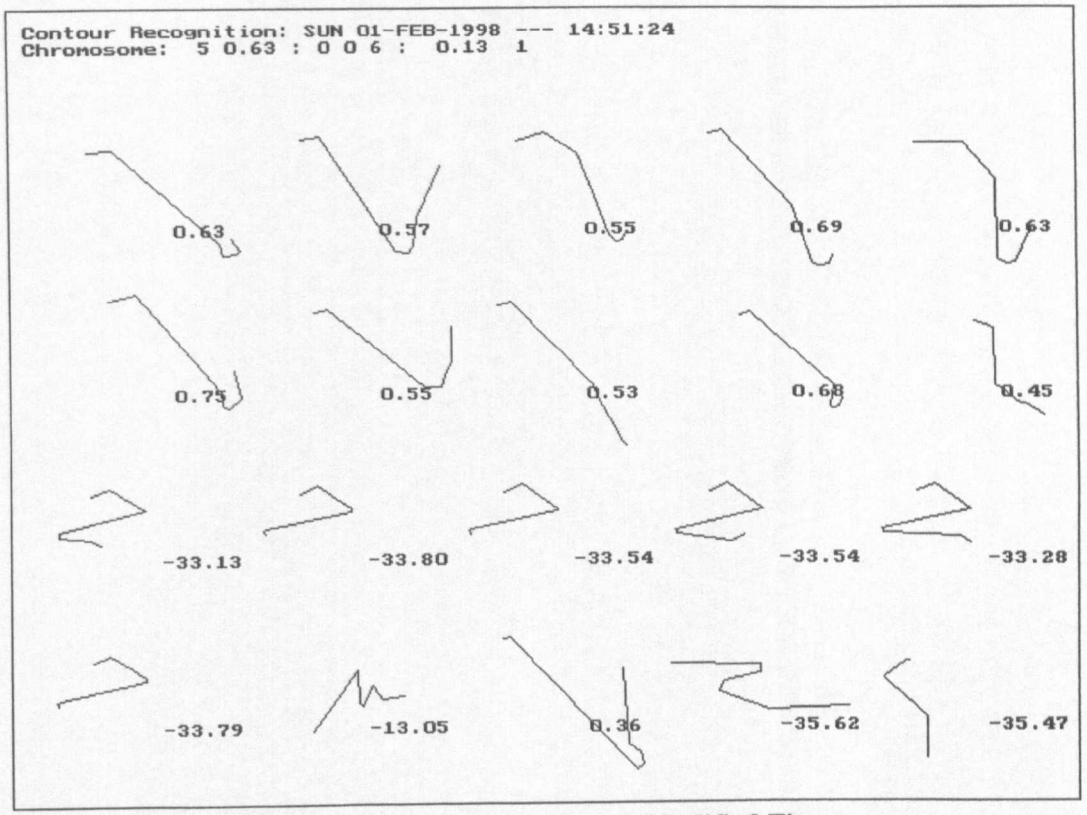

Figure 7-21: Recognition using Chromosome 5 with Modified Fitness

7.5 Summary and Conclusions

The research experiments described in this chapter have demonstrated various advantages and disadvantages of the fitness function calculations examined, and the results are summarised as follows:

1. A local incorrect maximum can occur in the fitness calculation that will direct the genetic algorithm to the wrong solutions of the problem (see Figure 7-13). This type of defect will make the fitness calculation unsuitable for use by the genetic algorithm.
2. A local minimum can occur in the fitness calculation that will bias the genetic algorithm away from chromosomes that may be potential contributors to a future solution (see Figure 7-14). The performance of the genetic algorithm will be reduced.
3. The fitness calculation can be asymmetric, with the variation of the fitness values either side of the best solution, having different slopes (see Figure 7-11). This defect will have to be considered when using a ranking method for the selection process. The range and slope of the

ranking function may have to be adjusted according to the *parameter values* that are associated with each fitness value. The ranking method will have to normalise the fitness values differently either side of the maximum fitness, in any particular generation, so that each chromosome can have a probability of selection that is correctly proportional to its relative 'linear' ranking. Further research would be required to investigate the design of a ranking function that uses the parameter values rather than the fitness values of the chromosome.

4. The fitness calculation can be non-linear over different ranges of the fitness (see Figure 7-12, Figure 7-13 and Figure 7-15). This characteristic will also have to be considered when using a ranking method for the selection process. The range and slope of the ranking function will, in this case, have to be adjusted according to the *fitness value* rather than according to a *parameter value* used in the calculation. The relative fitness values can then be normalised over the various non-linear regions of the fitness calculations.
5. The triangular fitness function used by the genetic algorithm developed in Chapter 5 and used for contour shape recognition in Chapter 6 can bias the fitness of chromosomes to a lower value if the difference in the line-segment vector directions is calculated using only a clockwise rotation. This bias varies according to the direction of the reference line-segment. The triangular fitness function is non-linear for certain configurations of line-segments (see Figure 7-15) but can be linear for other combinations of line-segments (see Figure 7-16 and Figure 7-17). This bias can be removed if the direction fitness is calculated in a clockwise and/or anticlockwise rotation, depending on the direction of the reference line-segment (see modulus (4) modification, Figure 7-2 and Figure 7-3).

The research experiments described in this chapter have shown that it is imperative that the form of the fitness function is analysed. The above defects can either be avoided or included into a ranking function to provide a probability of selection that is proportional to the correct relative fitness of each chromosome. The schema theorem will then apply to the selection operators used by the

standard genetic algorithm in the correct manner and the standard genetic algorithm will then behave as predicted by the schema theorem. The modulus (4) modification to the triangular fitness function has been shown to be a satisfactory refinement to the fitness calculation for line-segment vector directions. Further research is required to show that the modified triangular fitness function is suitable for the correct calculation of line-segment vector directions for any two-dimensional shape boundary contour.

8. Schema and Fitness Analysis

8.1 Introduction

The research work described in this chapter analysed the behaviour of the various schema in the shape contour observation chromosome, as developed in Chapter 5, when the crossover and mutation probabilities were varied. This research aimed to show that the schema theorem applied to the genetic algorithm developed in Chapter 5 and that this genetic algorithm followed the behaviour of a standard or canonical genetic algorithm. The behaviour of the schema as the values of the crossover and mutation probabilities were adjusted to increase the diversity of the population (so that a number of less fit but 'good' chromosomes were evolved) was particularly examined. The analysis described below investigated the relative effect of crossover and mutation in discovering various less fit but 'good' chromosomes. This analysis is based on the work of Holland (1992) and Goldberg (1989). A possible explanation is given for the high diversity in the chromosomes for mutation probabilities greater than 0.2.

The fitness distributions of the fittest chromosomes that evolved for the contours of the digits two and four were also analysed. The results of this part of the research work suggested that the 'epistasis' of a chromosome should be obtained from a measure of the interaction of the 'genes' in the chromosome rather than any interaction between the 'bits' in the chromosome representation as described in Davidor (1991). An examination of the fitness distribution for all possible values for the chromosomes ('Grand Population') indicated that the genetic algorithm is to be preferred to a 'hill climbing' technique, especially when more than one solution is required.

An analysis of schema can suffer from data explosion effects, where the recording of the behaviour of various schema becomes prohibitive due to the large number of schema that are present in a chromosome. In order to reduce this data explosion, three particular schemas were observed. The schema (genes) chosen for analysis were the start quadrant schema (gene 1- two bits), the clockwise

schema (gene 2 - one bit) and the number of line segment schema (gene 3 - three bits), see chromosome structure in Chapter 5. The number of each of the schemas was recorded on disc, for each generation, together with the average fitness of the chromosome of which they are a part. The chromosomes with fitness greater than a pre-set value were recorded for each generation. The growth of the schema in the group of chromosomes that were evolved during the number of generations of the genetic algorithm was then analysed from this recorded data.

The following experiments were performed as part of the research work:

1. The crossover and mutation probabilities were first set to zero to observe the effects due to reproduction alone (Figure 8-1, Test 1).
2. The crossover probability was then varied with the mutation probability equal to zero (Figure 8-1, Test 2 to Test 6). The effects of crossover alone were then recorded and analysed.
3. The mutation probability was then varied with the crossover probability set to zero (Figure 8-1, Test 7 to Test 21). The effects of mutation alone were then recorded and analysed.
4. A random genetic algorithm was tested (Figure 8-1, Test 22).
5. Finally the crossover and mutation probabilities were set to non-zero values (Figure 8-1, Test 23 to Test 36) and the effects of the various probability values were recorded and analysed.

The importance of the relative values of the crossover and mutation probabilities in providing the correct diversity level in the population for the discovery of a group of less fit solutions was also investigated. The fitness distribution for all possible chromosome values (65536) was obtained in this research by an adaptation of the genetic algorithm that replaces the reproduction function with a function that provides successive values for the chromosome population as the generations proceed. The fitness value for each chromosome was recorded to a disc file. This operation took approximately two hours on a personal computer using a 20 MHz Intel 386 processor with a maths coprocessor. The distribution of fitness for various combinations of genes was analysed in order to discover if the structure of the genes in the chromosome was suitable for an efficient genetic

Test	Pc	Pm	Diversity	Maximum Fitness	Average Fitness	Unique Solutions	Different Solutions
1	0.0	0.0	7.94	0.55	0.55	3	3
2	0.1	0.0	7.94	0.60	0.60	7	4
3	0.3	0.0	7.94	0.67	0.67	4	4
4	0.6	0.0	3.29	0.81	0.49	32	8
5	0.8	0.0	3.11	0.81	0.44	57	13
6	1.0	0.0	4.82	0.81	0.48	44	6
7	0.0	0.005	7.94	0.64	0.64	8	2
8	0.0	0.01	6.30	0.64	0.57	16	5
9	0.0	0.10	2.60	0.81	0.36	84	11
10	0.0	0.20	1.56	0.67	0.26	27	12
11	0.0	0.30	1.62	0.81	0.26	40	15
12	0.0	0.40	1.02	0.67	0.19	21	15
13	0.0	0.50	1.12	0.64	0.19	12	10
14	0.0	0.60	1.16	0.81	0.18	15	8
15	0.0	0.70	1.25	0.81	0.20	21	13
16	0.0	0.80	1.16	0.67	0.19	15	10
17	0.0	0.90	1.35	0.81	0.18	16	9
18	0.0	0.95	1.46	0.63	0.20	13	10
19	0.0	0.97	3.10	0.81	0.43	32	6
20	0.0	0.98	2.65	0.67	0.28	11	3
21	0.0	0.99	5.37	0.64	0.42	11	3
22	Random	GA	0.89	0.58	0.17	9	5
23	0.6	0.1	2.88	0.64	0.31	20	6
24	0.7	0.1	1.72	0.81	0.33	39	16
25	0.8	0.1	2.35	0.81	0.33	35	10
26	0.9	0.1	1.74	0.67	0.26	21	11
27	0.6	0.2	1.44	0.67	0.22	22	12
28	0.7	0.2	1.29	0.63	0.20	20	10
29	0.8	0.2	1.49	0.81	0.24	20	13
30	0.9	0.2	1.39	0.64	0.21	20	7
31	0.6	0.3	1.05	0.81	0.23	23	12
32	0.7	0.3	1.31	0.64	0.24	23	11
33	0.8	0.3	1.54	0.61	0.22	21	11
34	0.9	0.3	1.73	0.81	0.26	25	9
35	1.0	1.0	3.27	0.67	0.32	25	6
36	0.0	1.0	7.02	0.59	0.16	2	2

Figure 8-1: Summary of Test Results for Various Values of Pc and Pm

algorithm application.

8.2 Schema Theory Review

The schema analysis, described in this section, is taken from Goldberg (1989, pages 28 to 33) and Holland (1992, Chapters 4 and 6). The analysis is reproduced from these two sources for completeness and is presented in a form that is useful for the analysis of the research experiments described in this chapter. A chromosome A is represented as a string of individual characters. The string is described in the form $A = a_1 a_2 \ldots a_i$. 'a_i' is referred to as a 'gene' that has a value formed from the binary alphabet $V = \{0,1\}$. The actual value of a gene, $val(a_i)$, is referred to as the 'alleles' of the gene. The position of the gene in the chromosome determines its function.

A number of individual strings $A_j, j = 1 \; to \; n$ are contained in the population $A(t)$, of size n, at generation t. A schema H is formed from the alphabet triplet $V^+ = \{0,1,*\}$, where * is a 'don't care' or 'wildcard' symbol. An example schema could be H = *11*0 **. For alphabets of cardinality k there are $(k+1)^l$ schema where l is the length of the string (chromosome). For a population of n members there are at the most $n.2^l$ different schema contained in the population (each chromosome represents 2^l schema). These simple calculations indicate the magnitude of the information being processed by a genetic algorithm.

All schema are not created equal. Some schema are more 'specific', i.e. have less '*' in the alleles. Schema also vary in length. Hence the various schema in a chromosome can be more or less specific and will vary in length. The 'order' of a schema, $o(H)$, is defined as the number of fixed positions, e.g. for the schema $H = 011*1**$, $o(H) = 4$. The 'defining length' of a schema, $\delta(H)$, is specified as the distance between the first and last specific bit position, e.g. for the schema H = 011*1**, $\delta(H)$ = (last - first) = (5 - 1) = 4.

8.2.1 Effect of Reproduction Alone

The effect of reproduction alone on the "expected" number of schema in a population can be analysed in the following way. At generation t, the m examples of a schema H contained in the population $A(t)$ are given by $m = m(H,t)$. During reproduction, a string (chromosome) is copied according to its 'fitness'. A string A_j is selected with probability $p_j = \frac{f_j}{\sum f_j}$, where $j = 1$ *to* n, with the size of the population = n. f_j = fitness of the string A_j and $\sum f_j$ = total fitness of the population. Hence, the string A_j in the population A is selected with a probability proportional to its relative fitness compared to the total fitness of the population. For a population of size n, with replacement from the population $A(t)$, we expect to have $m(H,t+1)$ representatives of the schema H in the population $A(t+1)$ given by the following equation:

$$m(H,t+1) \geq \frac{m(H,t)\, n\, f(H)}{\sum f_j} \tag{8.1}$$

where $f(H)$ is now the average fitness of the string that contains schema H at generation (t).

The average fitness of the of the whole population $\bar{f}$ is given by:

$$\bar{f} = \frac{1}{n}\sum f_j \tag{8.2}$$

Therefore we have:

$$m(H,t+1) \geq \frac{m(H,t) f(H)}{\bar{f}} \tag{8.3}$$

A particular schema, therefore, grows (or decays) as the ratio of its 'average' fitness to the 'average'

fitness of the whole population. Schema in a chromosome with fitness values greater than the population average will receive an increasing number of samples in the next generation. Likewise schema in a chromosome with fitness values less than the population average will receive a decreasing number of samples in the next generation. Note that this 'expected' behaviour is carried out with every schema H contained in the population A in 'parallel'. Holland (1992, page 103) states that "Theorem 6.2.3 provides the first evidence of the intrinsic parallelism of genetic plans. **Each** schema represented in the population increases or decreases according to the above formulation **independently of what is happening to the other schema** in the population. The proportion of each schema is essentially determined by its average performance in relation to the population average."

Re-arranging equation (8.3) above, we have:

$$m(H,t+1) \geq k_f m(H,t) \tag{8.4}$$

$$\text{where } k_f = \frac{f(H)}{\bar{f}} \tag{8.5}$$

Hence we can write:

$$m(H,t) \geq m(H,0) k_f{}^t \tag{8.6}$$

Equation (8.6) is in the form of a geometric progression or exponential form. Goldberg (1989, page 30) assumed a stationary value for k_f, whereas Holland (1992, pages 87, 125 and 181) mentioned a time varying value for k_f. It is important to note that, in general, k_f is a function of the generation value t, i.e. $k_f = k_f(t)$ and is not a constant (see also Section 8.3).

The effect of reproduction 'alone' allocates an exponentially increasing (decreasing) number of trials to above (below) average schema and in parallel. Note, also, that reproduction alone 'exploits' the schema in the population but does nothing to promote 'exploration' of new regions of the schema space. The reproduction only exploits the original random selection of chromosomes and does not provide any new combinations of genes in the population.

8.2.2 Effect of Crossover Alone

The effect of crossover alone on the 'expected' number of schema in a population can be analysed in the following way. A crossover site is selected uniformly at random. If the length of the chromosome equals l, then the possible number of sites equals $(l-1)$. The probability of destruction, p_d, is given by:

$$p_d = \delta(H)/(l-1) \tag{8.7}$$

Because a schema survives when the crossover site falls outside the defining length, the probability of survival, p_s, is given by $p_s = 1 - p_d$, since the schema is likely to be disrupted whenever a site within the defining length is selected from the $(l-1)$ possible sites. If the crossover itself has a probability p_c, then the survival probability, k_c, may be given by the expression:

$$k_c \geq 1 - p_c\delta(H)/(l-1) \tag{8.8}$$

The value of $\delta(H)$ for a single bit schema equals zero. Hence k_c equals 1.0, and the crossover operator will not affect the single bit schema. For p_c = 1.0, we have:

$$k_c = 1 - \delta(H)/(l-1) \tag{8.9}$$

The effect of reproduction and crossover together on the 'expected' number of schema in a

population is described below.

Adding k_c to equation (8.4) we have:

$$m(H,t+1) \geq k_f k_c m(H,t) \tag{8.10}$$

Thus schema H grows or decays depending on a multiplication factor.

This multiplication factor depends on the following parameters (see Goldberg, 1989):

1. Is the average fitness of the schema above or below the population average fitness? This factor is contained in the parameter k_f.
2. Has the schema a relatively short or long defining length? This factor is contained in the parameter k_c.

Hence schema, with above average fitness and with short defining lengths, are going to be sampled at exponentially increasing rates and in parallel. This parallelism is noted in Holland (1992, page 104) where he states that "Crossing-over serves the adaptive process by continually introducing new schema for trial, while testing extant schema in new contexts – all this without disturbing the ranking process (except for longer schema)". Holland (1992, page 127) also makes the point that "... even though plans try *structures* from the population of chromosomes one at a time, it is really *schema* which are being tested and ranked. There are somewhere between 2^l and $n.2^l$ schema with instances in the population. *Each* one changes its proportion in the population at a rate largely determined by *its* observed performance, and is largely uninfluenced by what is happening to other schema. This is the foundation of the intrinsic parallelism of the genetic plans".

8.2.3 Effect of Mutation Alone

The effect of mutation alone on the 'expected' number of schema in a population can also be

analysed in a similar manner. Mutation is a random alteration of a single position in the chromosome with probability p_m. For a schema H to survive, all specified positions must survive. A single position will, therefore, survive with a probability of $(1 - p_m)$. Each mutation is statistically independent. If $o(H)$ = the number of fixed positions within the schema, then the probability of surviving the mutation, k_m, is given by the expression:

$$k_m = (1 - p_m)^{o(H)} \tag{8.11}$$

Adding k_m to equation (8.10), we can now write:

$$m(H, t+1) \geq k_f k_c k_m m(H, t) \tag{8.12}$$

Thus the growth or decay of schema H now has the additional multiplication factor due to the mutation process. The more fixed positions that are present in a schema the more that schema will be disrupted by the mutation operator.

8.2.4 Schema Theorem

Let $$k_s = k_f k_c k_m \tag{8.13}$$

Therefore $$m(H, t+1) = k_s m(H, t) \tag{8.14}$$

The above analysis shows that short, low-order and above-average schemas receive exponentially increasing number of trials in subsequent generations. This states the schema theorem or the fundamental theorem of genetic algorithms. Holland (1992, page 140) gives the following general summary of his schema theorem, "Schema of above-average performance are combined and tested in new contexts by crossing-over outside their defining locations. Because (the instances of) schema increase or decrease exponentially in terms of observed performance, the overall average

performance is close to the best observed. Because a wide range of promising variants is generated and tested entrapment on 'false peaks' (local optima) is prevented. Even for moderate sizes of population and representation, if the initial population is varied, a crossover probability > 0.5 will make it almost certain that every structure generated during the initial stages of adaption is **new**. Nevertheless, this high crossover probability does **not** disturb the rankings of schema that are consistently above average."

8.3 Discussion of Schema Analysis Test Results

The research performed various schema analysis experimental tests for a sixteen-bit chromosome. The genetic algorithm developed in Chapter 5 was used for these tests with another different set of crossover and mutation probabilities. The genetic algorithm code was modified to collect the schema (gene) statistics and record the appropriate data onto disc file. The results of these tests are summarised in Figure 8-1. The fitness calculation 'number of standard deviations' was set to 1.0 (see Chapter 5) and a fitness threshold set to 0.4, when storing the line-segment information onto disc for each generation. Note that these experimental tests were *completely separate* from those performed in Chapter 5.

Test 2 to Test 6 were concerned with variations in the crossover probability only. Test 7 to Test 21 examined the effects of changes in the mutation probability alone. Test 22 considered a random selection of chromosomes that did not involve the use of the crossover and mutation operators. Test 23 to Test 36 considered a mixture of crossover and mutation probabilities.

Test 1 showed the effect of the crossover and mutation probabilities being equal to 0.0, i.e. the effect of selection by itself. A single solution was eventually evolved, after 14 generations, of fitness 0.55 and a small number of unique and different solutions were developed. This single solution had a chromosome of the form [0-1-4 0.13 3] (see Chapter 5 for the specification of the chromosome). Figure 8-2, Figure 8-3 and Figure 8-4 show graphs of the actual and expected values for the number

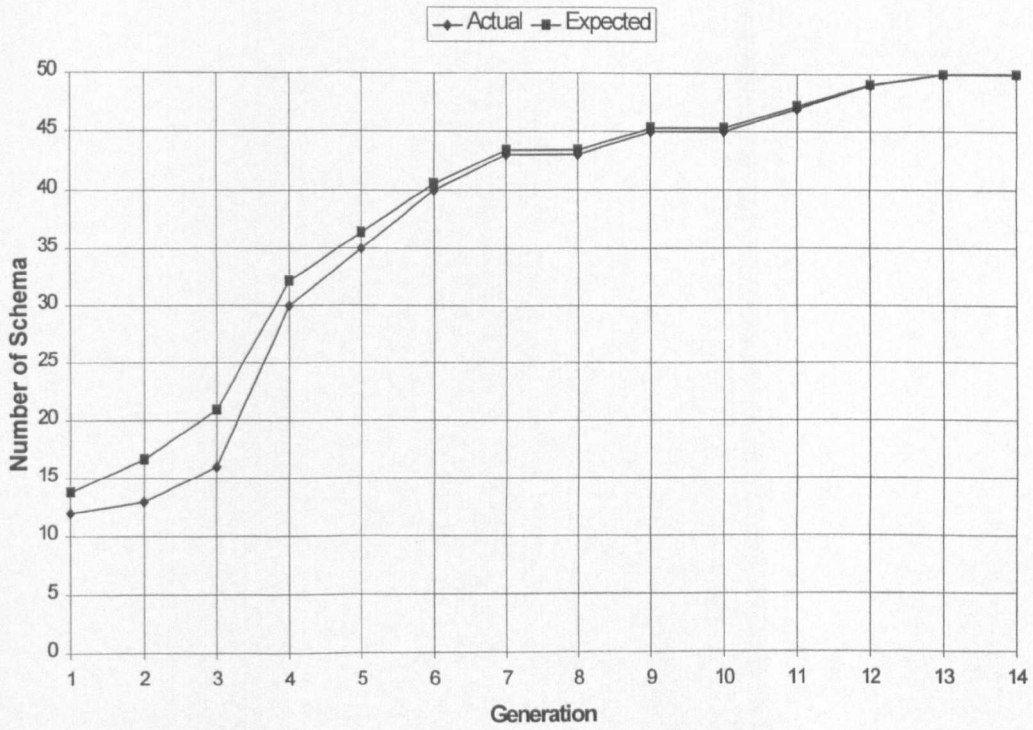

Figure 8-2: Number of Start Quadrant Schema [0] for Pc = 0.0 & Pm = 0.0

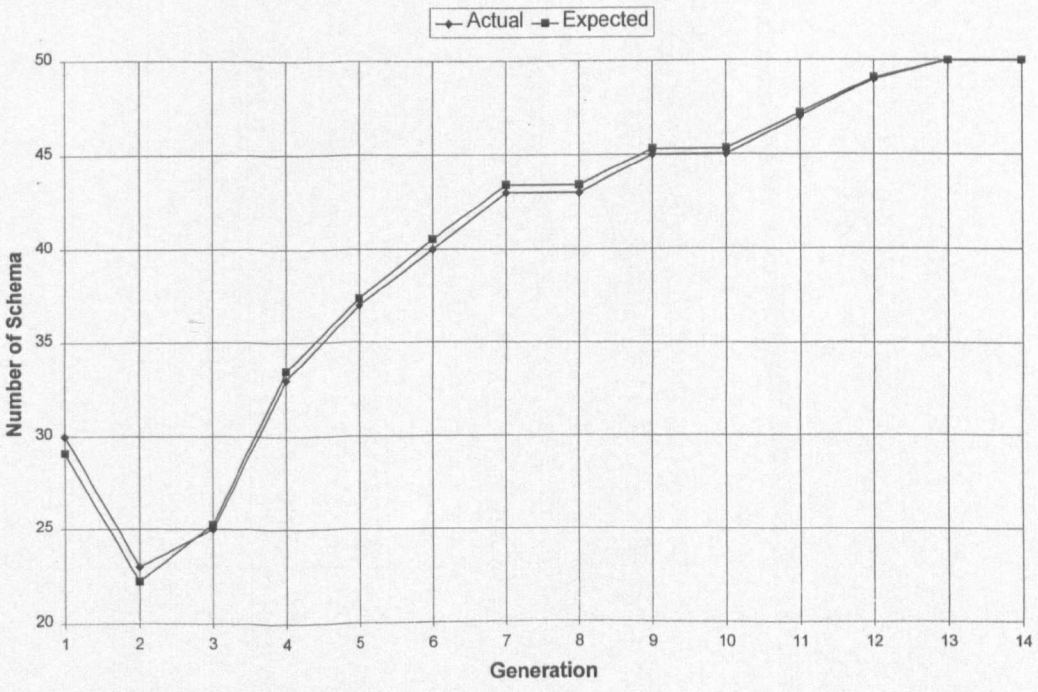

Figure 8-3: Number of Clockwise/Anticlockwise Schema [1] for Pc = 0.0 & Pm = 0.0

of schema at each generation, where the schema are 0, 1 and 4 respectively (i.e. observation genes 1, 2 and 3) and are from the single solution chromosome mentioned above.

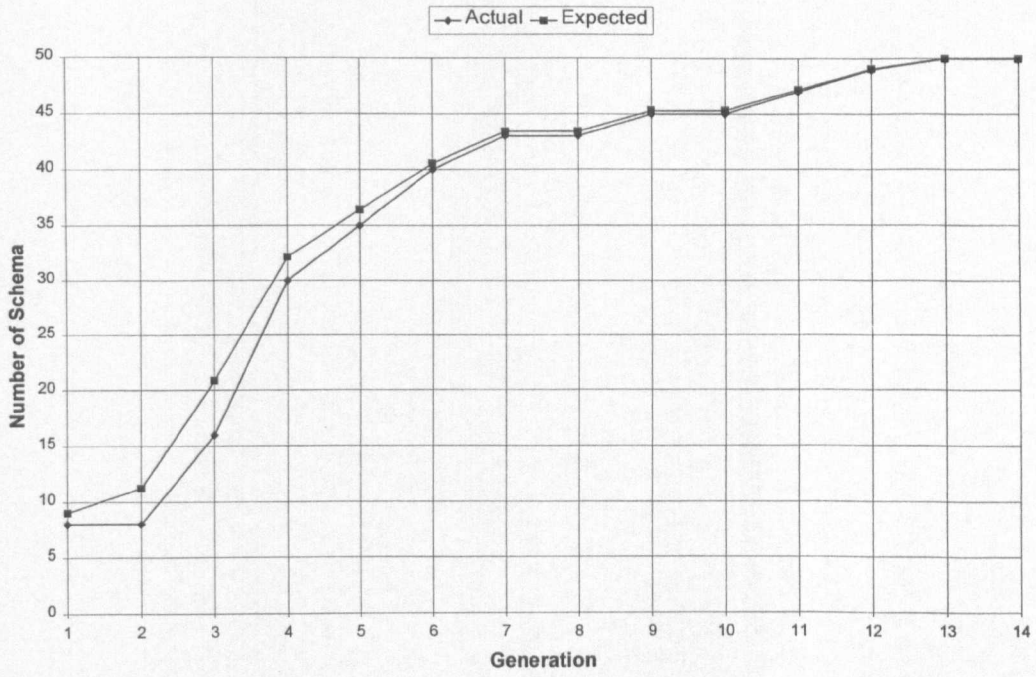

Figure 8-4: Number of Line-Segments Schema [4] for Pc = 0.0 & Pm = 0.0

These graphs show an 'exponential-like' increase of the number of schema, the rate of which varies with time. Figure 8-5, Figure 8-6 and Figure 8-7 show the variation in K_s over the 14 generations. The value of K_s is shown to vary with time and often increases in value during the initial generations of the genetic algorithm and gradually reduces to the value 1.0 as the generations proceed. These results indicate that the 'roulette wheel' selection used for reproduction in the genetic algorithm in Chapter 5 follows the theory discussed above, and that the rate of 'exponential-like' growth does vary as the generations proceed.

Test 2 to Test 6 showed the behaviour of the genetic algorithm for non-zero crossover probabilities only. The diversity measure indicated a low diversity until the crossover probability had values in the range 0.6 to 0.8, where the diversity could be considered to be of a 'medium' value (diversity measure about 3.0). The number of unique and different solutions also showed a large increase over this range of crossover probabilities. Higher values of the crossover probabilities appeared to reduce the number of different solutions. The lower crossover probabilities (< 0.6) helped the selection

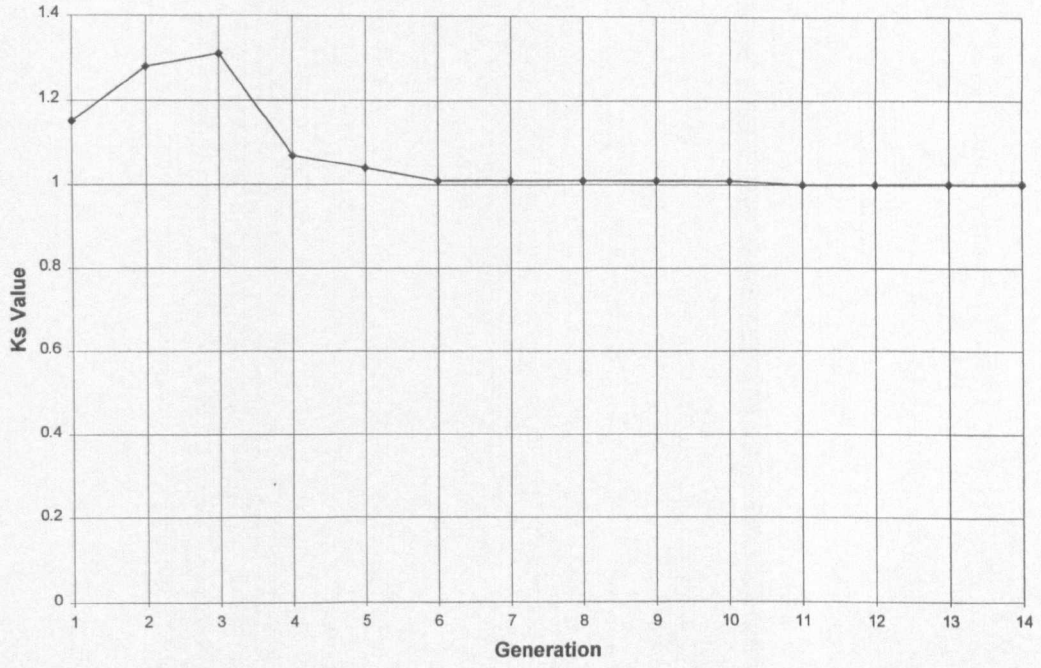

Figure 8-5: Ks Values for Start Quadrant Schema [0] for Pc = 0.0 & Pm = 0.0

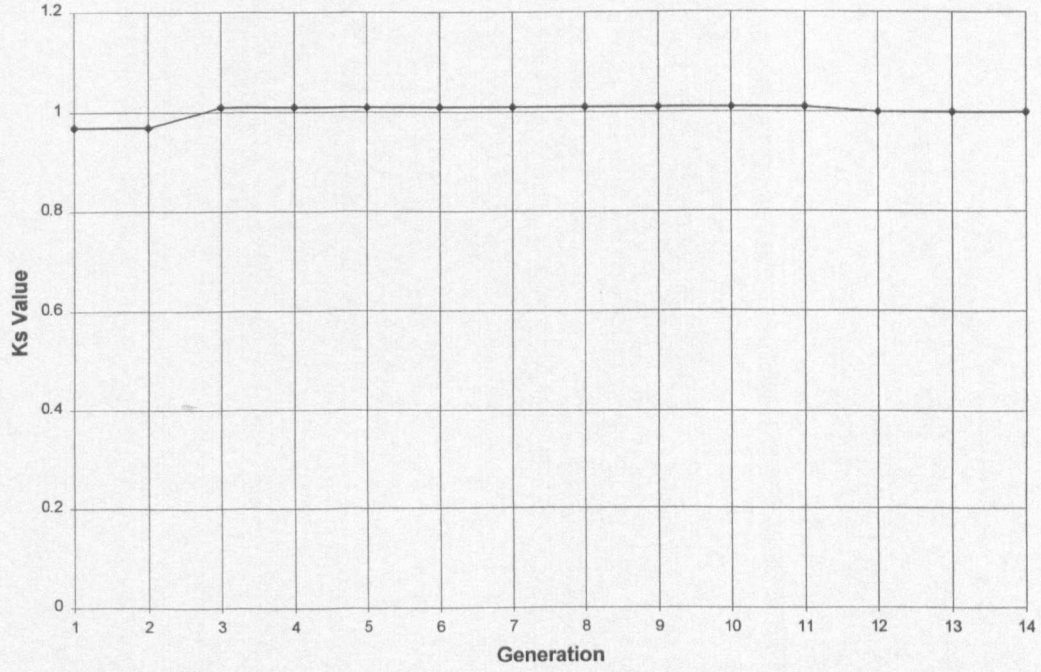

Figure 8-6: Ks Values for Clockwise/Anticlockwise Schema [1] for Pc = 0.0 & Pm = 0.0

process in finding a smaller number of solutions but not necessarily the best solution (fitness = 0.81). Crossover probabilities ≥ 0.6 also found the best solution and at the same time increased the diversity of the population.

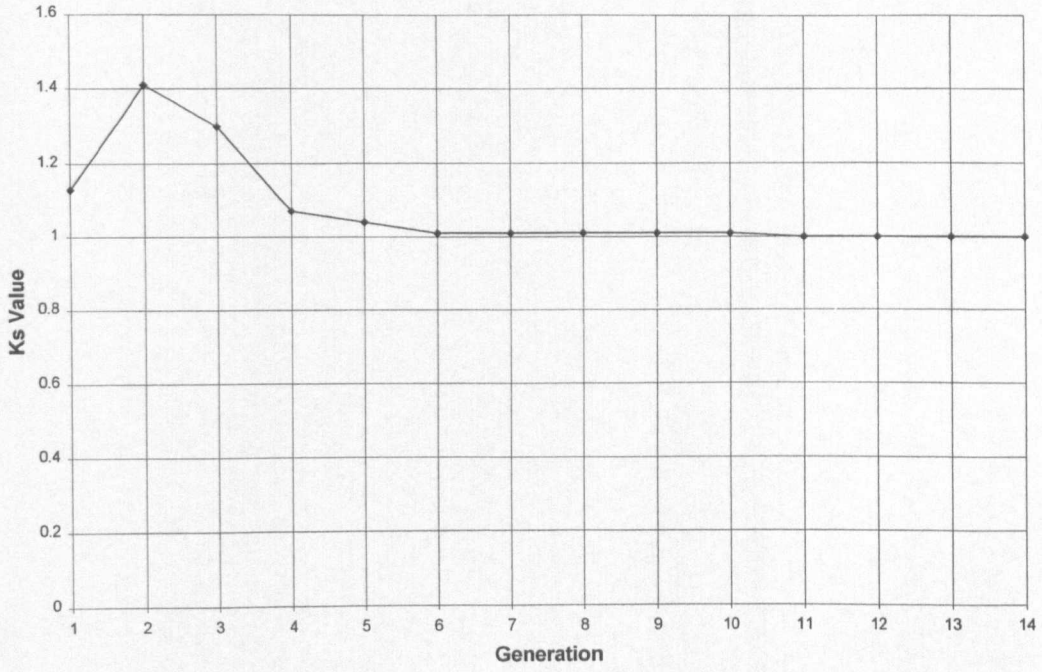

Figure 8-7: Ks Values for Number of Line-Segments Schema [4] for Pc = 0.0 & Pm = 0.0

Figure 8-8 to Figure 8-11 show the development of the gene combination 0-0-5 during Test 5. This gene combination with a fitness value of 0.63 gave 100% recognition of the digit two. The K_s values are generally > 1.0 and with the strong influence of the crossover operator, the start quadrant schema [0] and the number of line-segments, schema [5] show rapid growth. The K_s values for the search direction [0] oscillate about the value 1.0 and thus show a lower growth rate and are more influenced by the selection process.

Test 7 to Test 21 showed the behaviour of the genetic algorithm for non-zero mutation probabilities only. Maximum diversity (diversity measure about 1.2) of the chromosome population occurred for mutation probabilities between 0.2 and 0.95. The population average fitness was also low (about 0.2) and the number of unique and different solutions remained relatively high.

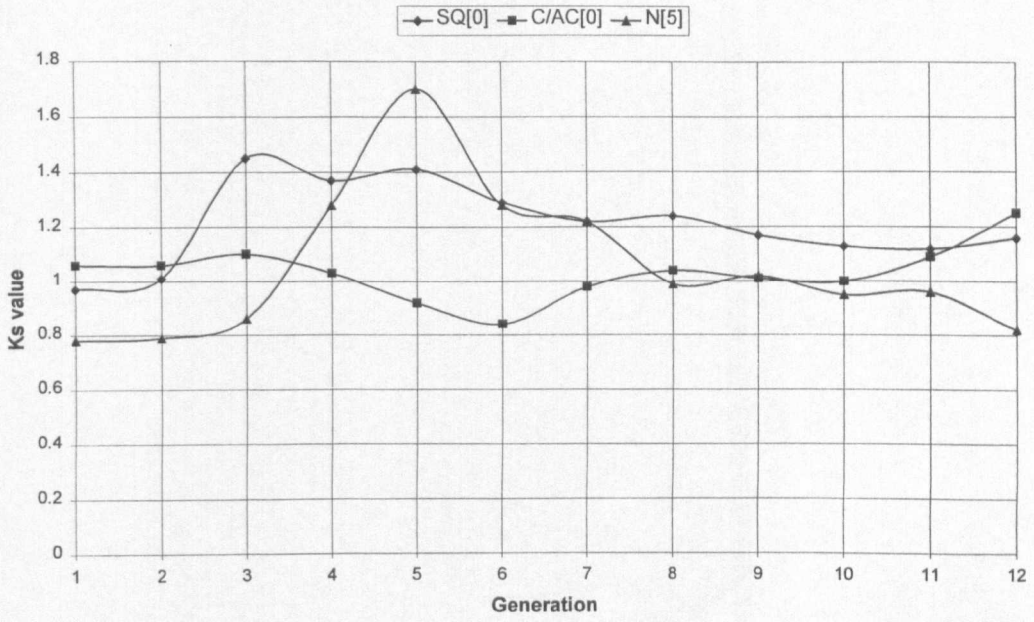

Figure 8-8: Test 5 : Ks Values : Pc = 0.8 Pm = 0.0

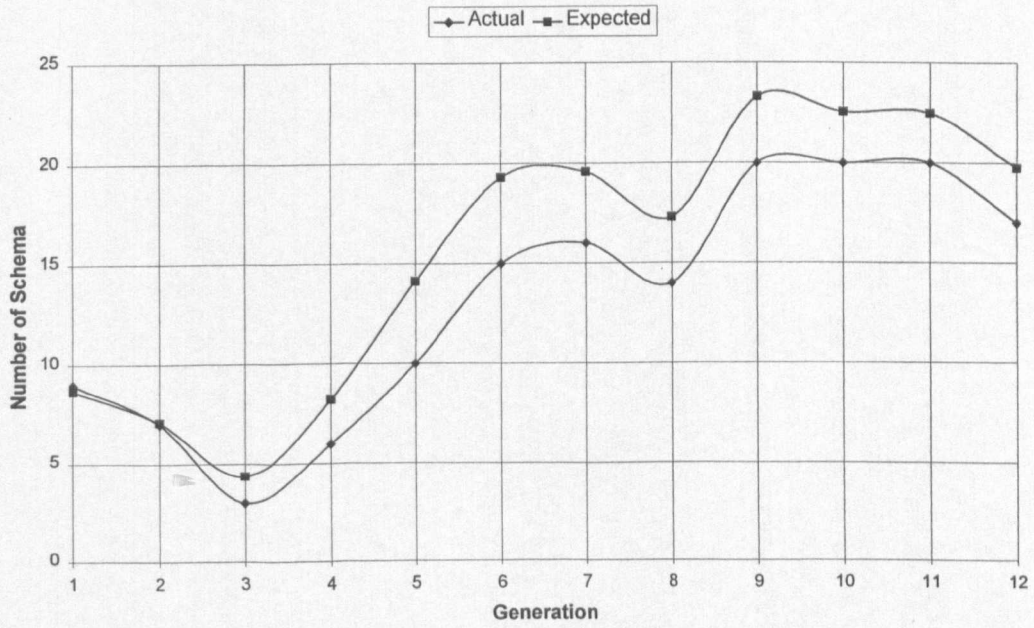

Figure 8-9: Start Quadrant Schema [0] : Pc = 0.8 Pm = 0.0

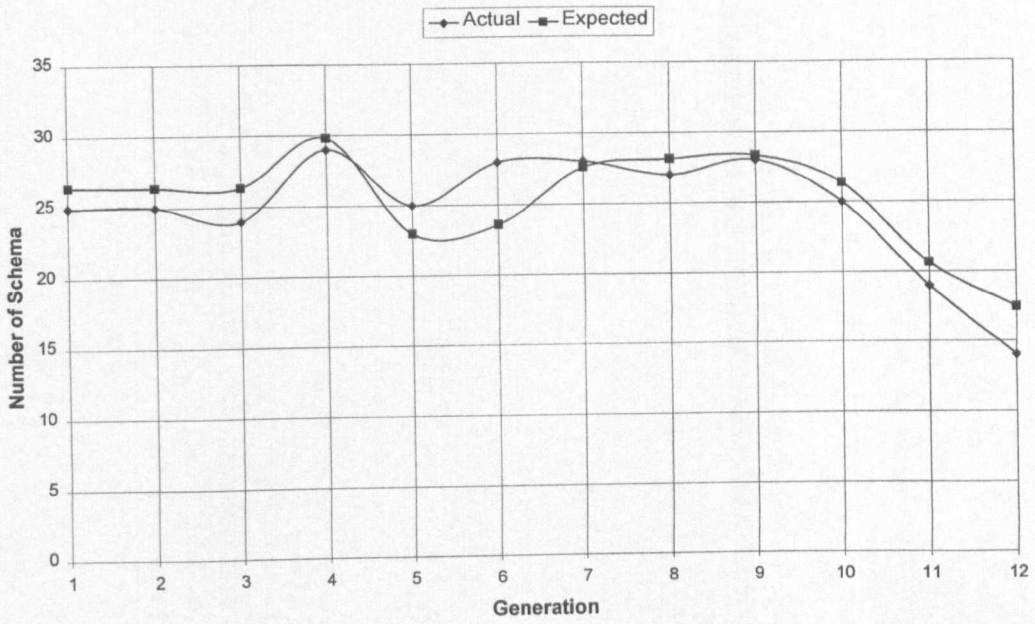

Figure 8-10: Clockwise/Anti-Clockwise Schema [0] : Pc = 0.8 Pm = 0.0

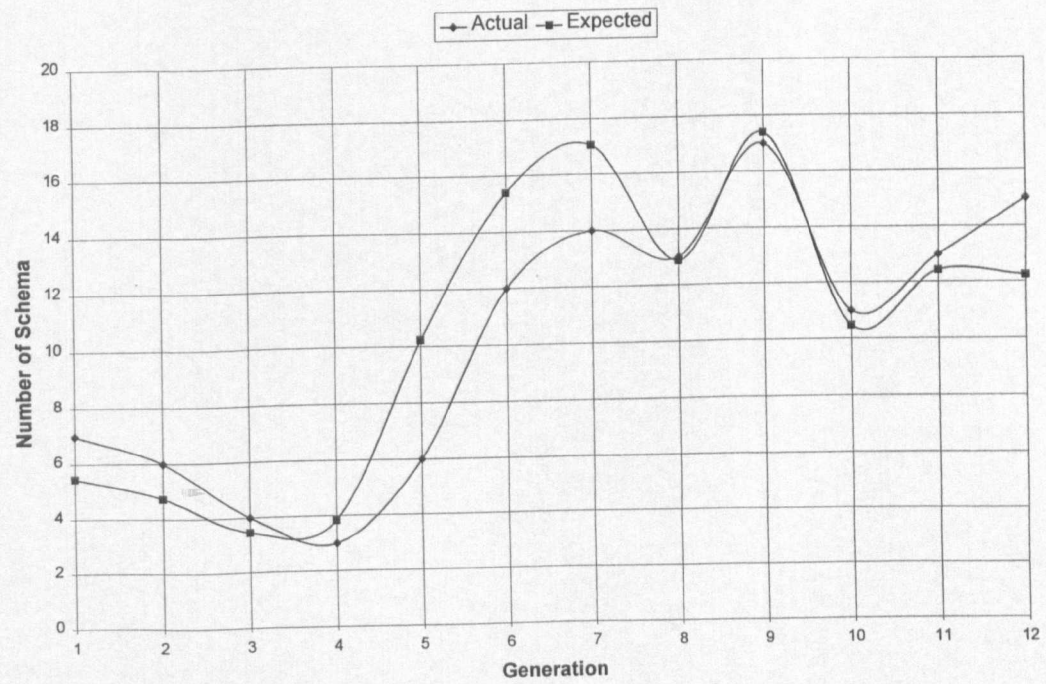

Figure 8-11: Number of Line-Segments Schema [5] : Pc = 0.8 Pm = 0.0

Note that, the genetic algorithm in Chapter 5 required a high value for the unique and different solutions so that a contour recognition capability could be achieved. If only one way to observe a contour was evolved, then other contours may have also give a high fitness value for this particular observation. A variety of ways to observe a contour would protect the recognition process from parts of different contours being similar, and thus have given a high fitness value for those parts of

the contours. More ways to observe a contour would generate more variety in the recognition process.

For mutation probabilities > 0.95 the diversity appeared to decrease and the number of appropriate solutions also decreased. The genetic algorithm also showed an oscillatory behaviour as the mutation probability was increased to the value 1.0. As more bits in the chromosome were 'changed' from the value 1 to 0 and vice versa, the fitness of the chromosome alternated between a relatively high value and a relatively low value. This oscillatory behaviour destroyed the 'exploration' capability of the genetic algorithm.

Figure 8-12 to Figure 8-15 show the development of the gene combination 3-0-2 during Test 11. This gene combination with fitness equal to 0.55 gave 100% recognition of the digit two. The K_s values were below 1.0 and oscillated under the influence of the mutation operator. The presence of this gene combination peaked between generations 4 and 5, with the start quadrant schema [3] gradually reduced in numbers as the evolution proceeds. The presence of the mutation operator appeared to cause the oscillatory behaviour of this gene combination. The genes were evolving on the rising part of the K_s oscillation and were suffering extinction on the falling side of the K_s oscillation.

Test 22 shows the results for a typical random genetic algorithm, in which the chromosomes were constructed on each generation by a random number generator. The population had maximum diversity (about 0.9), with a reasonably fit best solution but low average fitness for the population as a whole. The number of unique and different solutions was also slightly low. The random choice of chromosome produced a highly diverse population but did not, in general, possess any of the exploration properties of the standard genetic algorithm. Figure 8-16 to Figure 8-19 show the development of the gene combination 3-0-8 during Test 22. This gene combination, with a fitness value of 0.48, gave 100% recognition of the digit two. The oscillatory nature of the random search

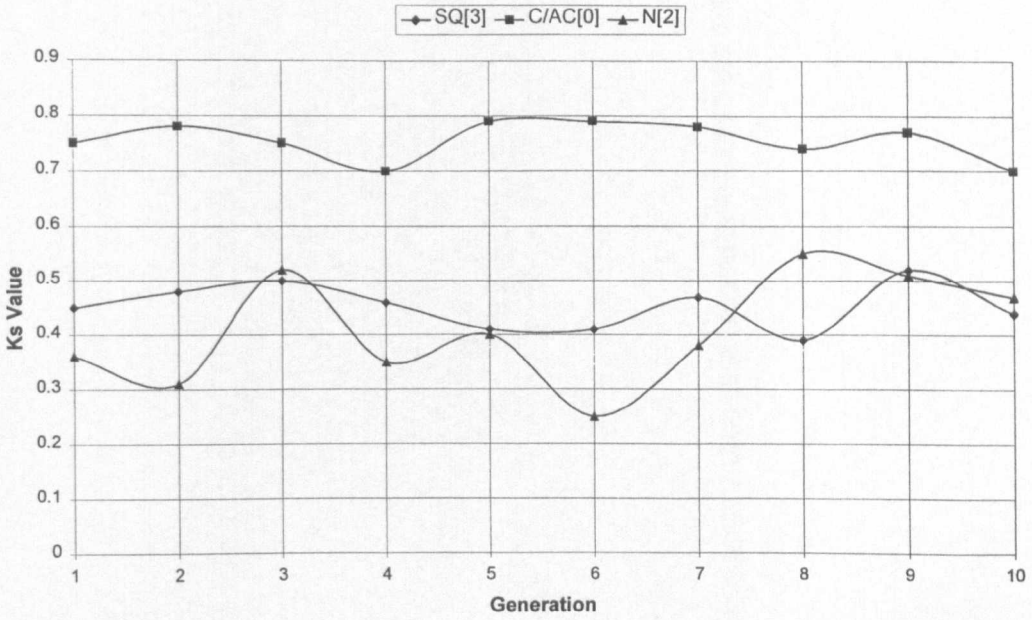

Figure 8-12: Test 11 : Ks Values : Pc = 0.0 Pm = 0.3

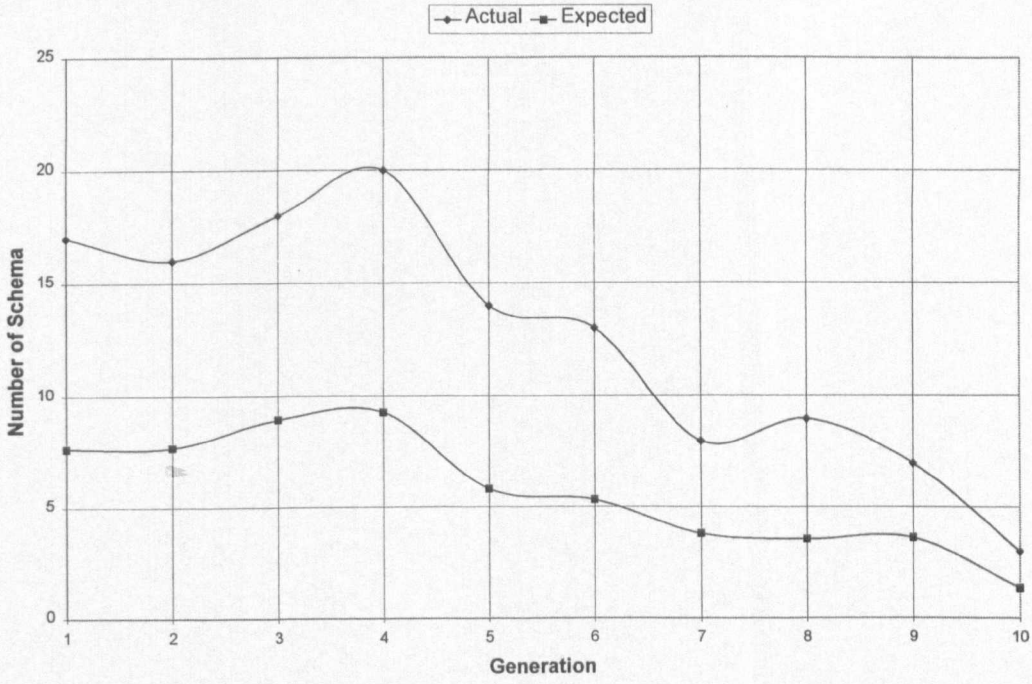

Figure 8-13: Start Quadrant Schema [3] : Pc = 0.0 Pm = 0.3

was evident with the values of K_s unable to be used to predict the development of the chromosome.

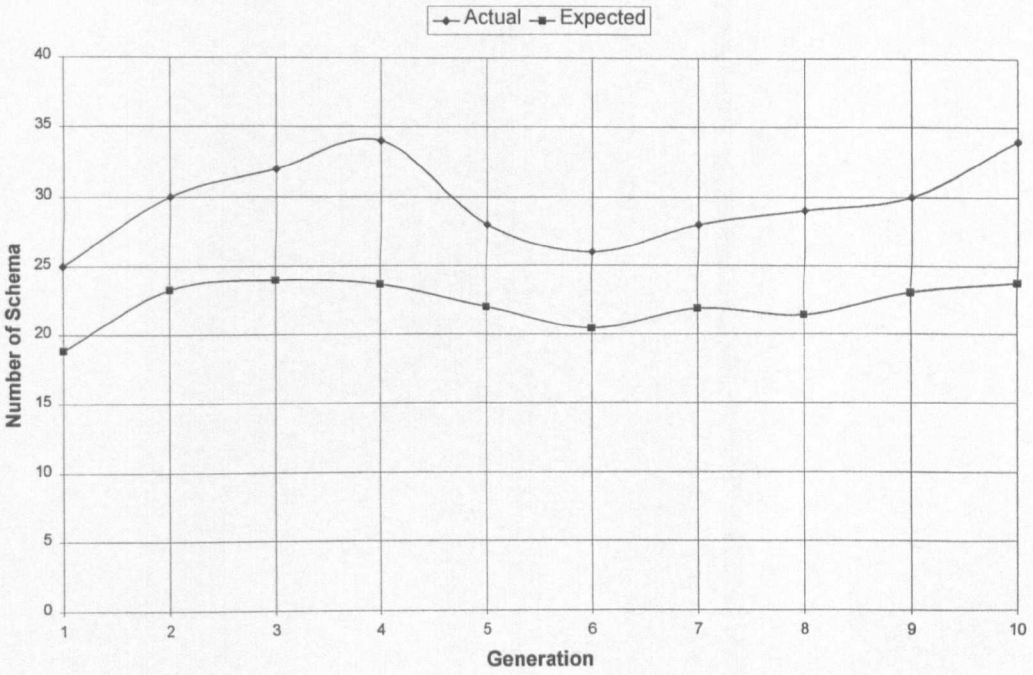

Figure 8-14: Clockwise/Anti-Clockwise Schema [0] : Pc = 0.0 Pm = 0.3

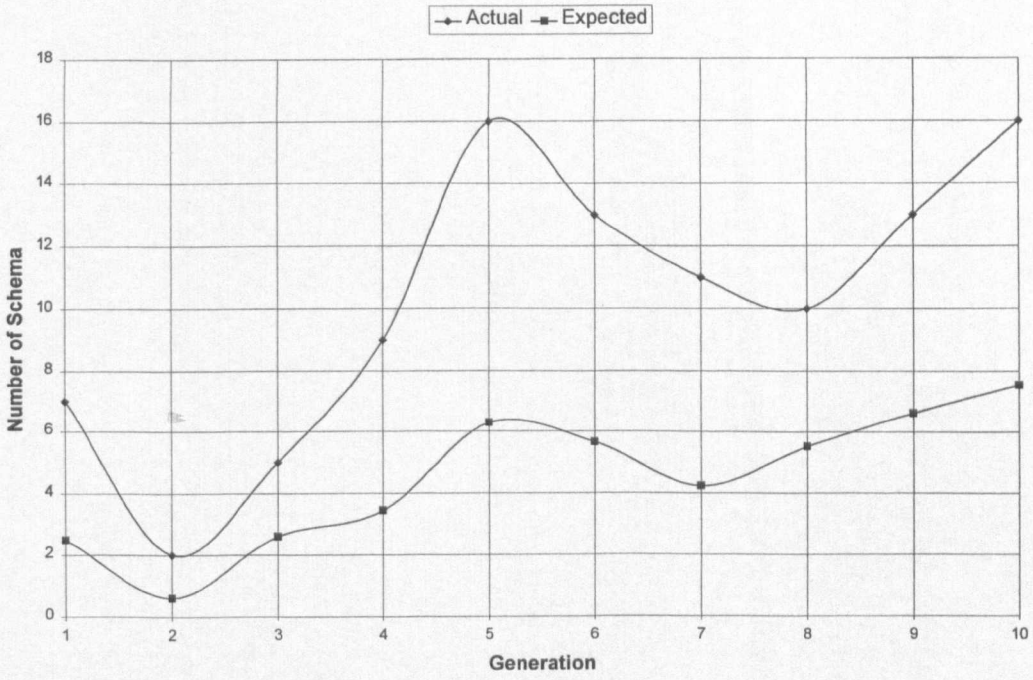

Figure 8-15: Number of Line-segments Schema [2] : Pc = 0.0 Pm = 0.3

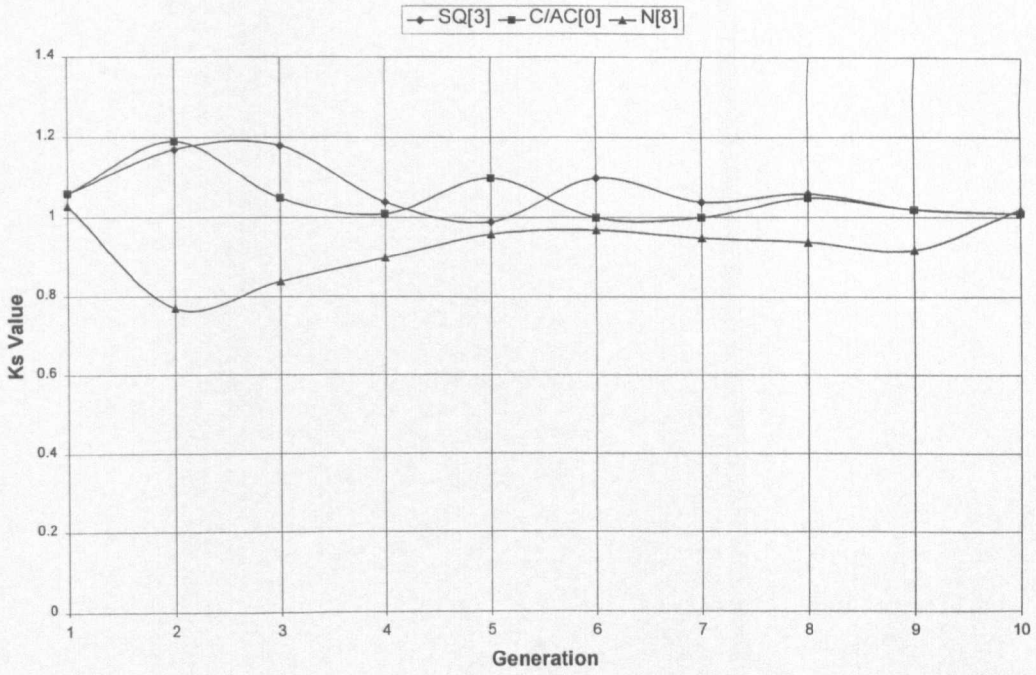

Figure 8-16: Test 22 Ks Values : Random Selection

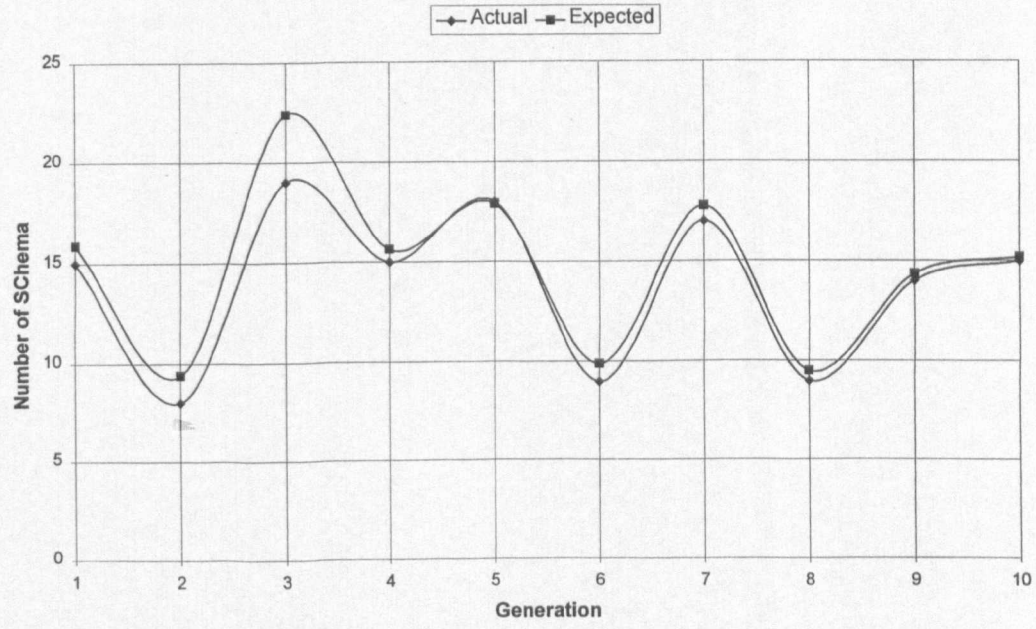

Figure 8-17: Start Quadrant Schema [3] : Random Selection

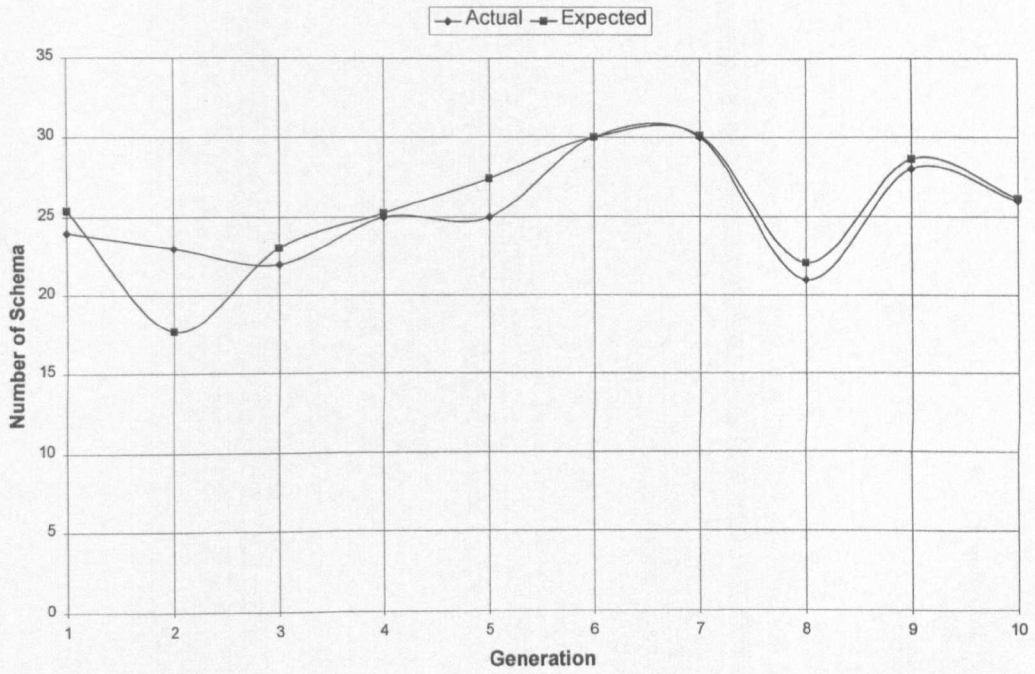

Figure 8-18: Clockwise/Anti-Clockwise Schema [0] : Random Selection

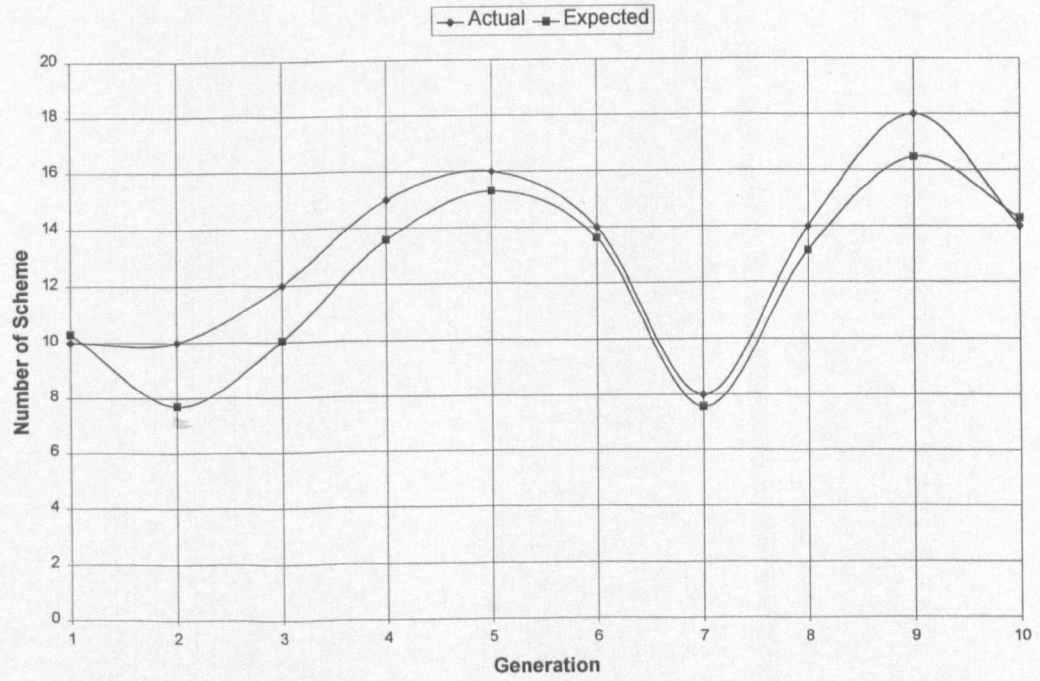

Figure 8-19: Number of Line-Segment Schema [8] : Random Selection

Test 23 to Test 34 provided results for a mixture of crossover and mutation probabilities. The crossover probability varied over the range 0.6 to 0.9 in order to give a reasonable diversity to the population (see also Test 2 to Test 6 above). The mutation probability varied over the range 0.1 to 0.3. The results showed that these ranges for the crossover and mutation probabilities provided a

high diversity in the population and a reasonably large number of unique solutions (about 20 to 30) and different solutions (about 10 to 15). The genetic algorithm (used in Chapter 5 to evolve various ways to observe a contour) should, therefore, use this range of crossover and mutation probabilities to obtain enough different ways of observing a contour so that recognition of the contour is possible.

Figure 8-20 to Figure 8-23 show the development of the gene combination 3-0-2 during Test 23. This gene combination, with a fitness value of 0.55, gave 100% recognition of the digit two. The K_s values remained above 0.8 after generation 4 with each gene generally showing an evolving increase in numbers as each generation progressed. Because of the high mutation probability, the value of K_s was below 1.0, showing the disturbance due to the mutation operator. The number of schema in this gene combination increased in value due to the crossover and selection operations. In this case the mutation operator provided the variation, i.e. higher diversity in the population.

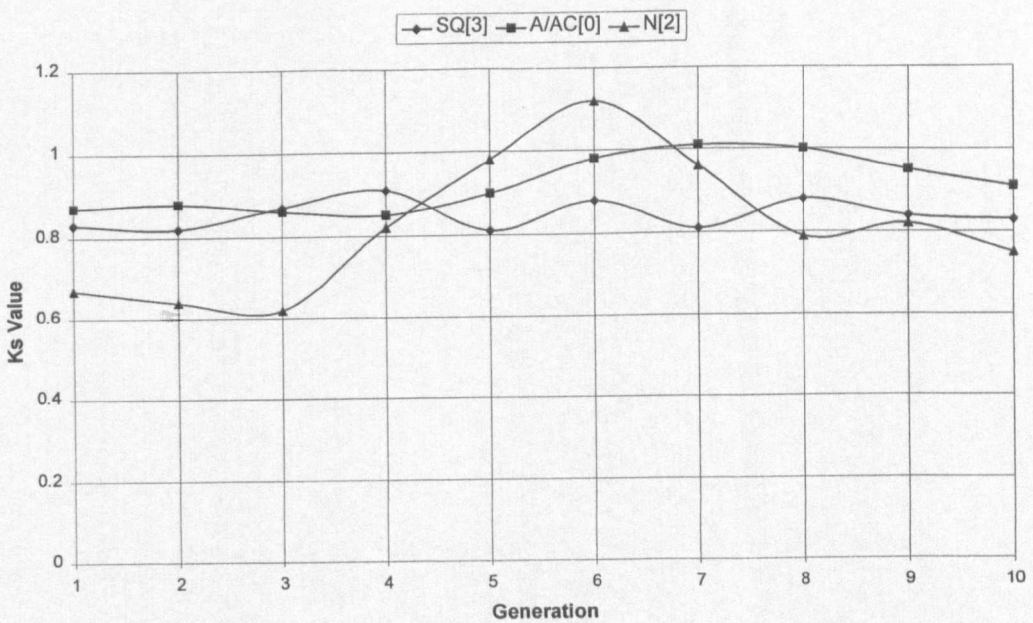

Figure 8-20: Test 23 : Ks Values : Pc = 0.6 Pm = 0.1

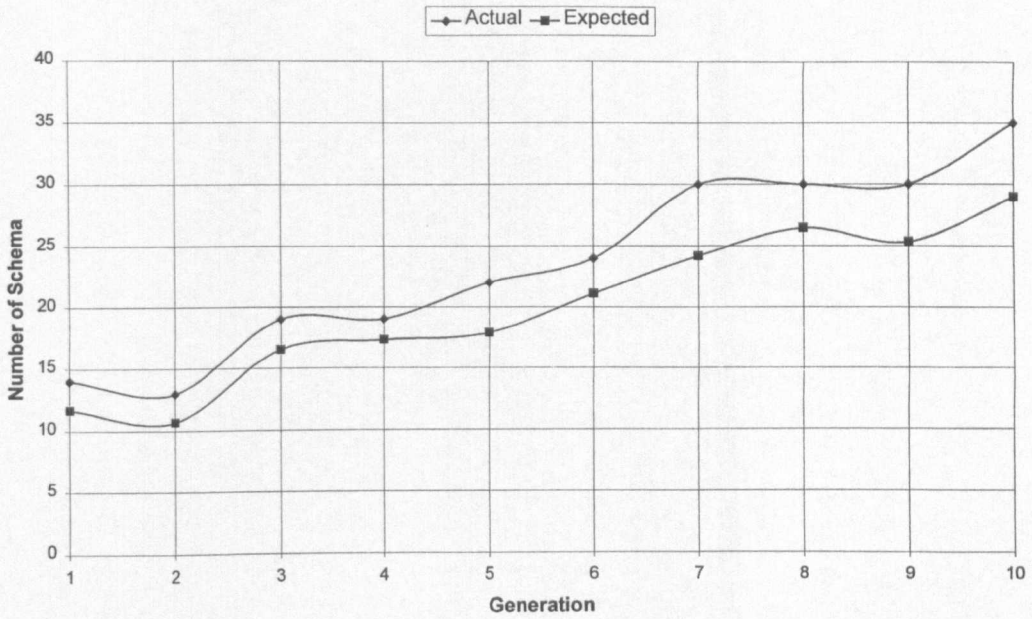

Figure 8-21: Start Quadrant Schema [3] : Pc = 0.6 Pm = 0.1

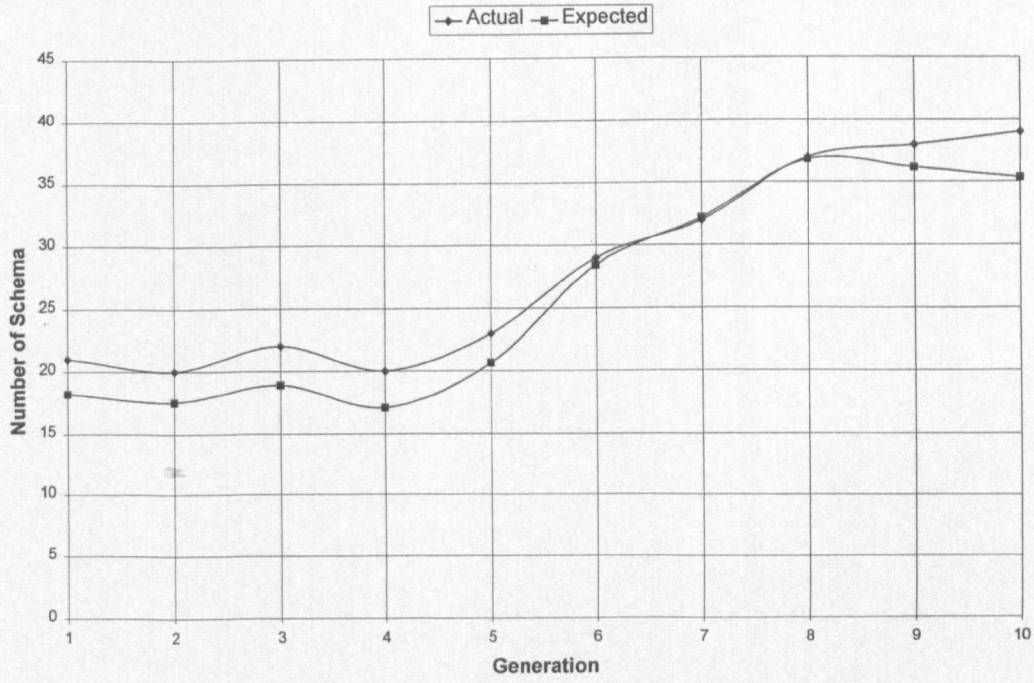

Figure 8-22: Clockwise/Anti-Clockwise Schema [0] : Pc = 0.6 Pm = 0.1

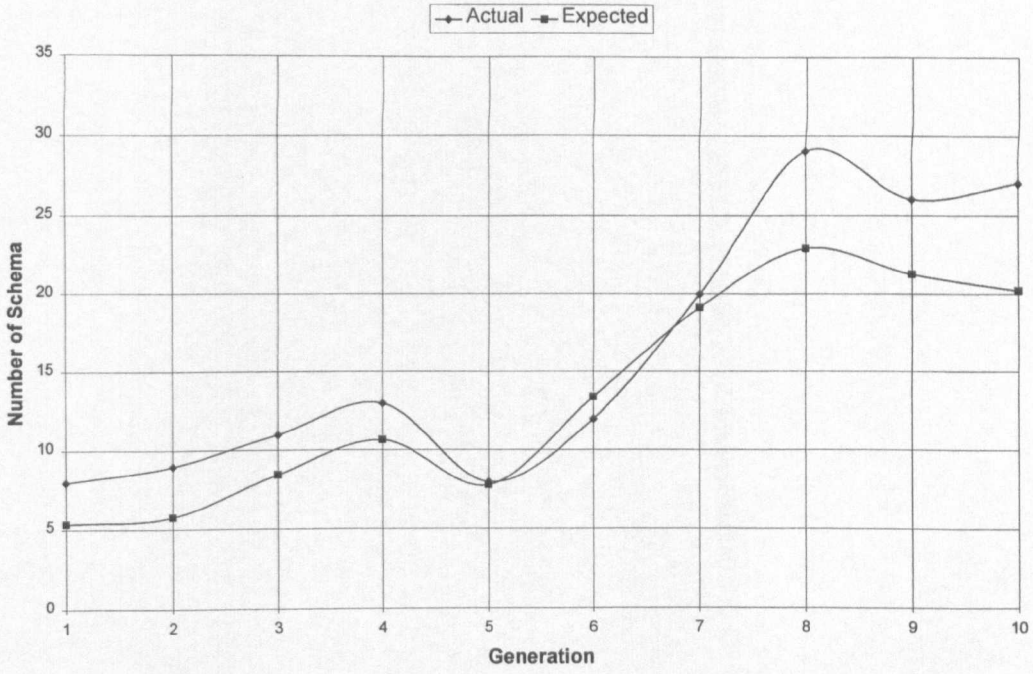

Figure 8-23: Number of Line-Segment Schema [2] : Pc = 0.6 Pm = 0.1

Test 35, with crossover and mutation probabilities equal to 1.0, showed the diversity decreasing to about 3.3 and the number of different solutions equalling 6. This effect was most likely due to the crossover probability of 1.0 causing a premature convergence towards too narrow a part of the solution space. Mutation was usually required to improve the diversity of the population, otherwise premature convergence would occur. Test 36 showed the lack of diversity in the population (about 7.0) and low values for the unique and different solutions due to a mutation probability of 1.0. The lack of diversity was due to the oscillatory effect that alternated the bits and produced a population that oscillated between high and low fitness on alternate generations. A mutation probability of 1.0 would 'change' each bit on every generation, and hence the population will only oscillate between two states, with the random initial highest fitness eventually predominating. Test 35, with the addition of the crossover probability of 1.0, improved the diversity of the population but did not provide as many different solutions. The high crossover probability counteracts the oscillatory behaviour of the high mutation probability by changing part of the chromosomes in the population before the bits are 'changed' by the mutation operator. Hence the population does not now oscillate between a high and a low fitness state, but can provide fitness values other than just the two

predominant values as in Test 36.

Figure 8-24 shows graphs for the values of K_c for different values of the crossover probability P_c. The graphs show the variation in K_c for schema of length 1, 2 and 3 respectively. For the chromosome used in the genetic algorithm in Chapter 5, a length 1 schema is the 'clockwise/anti-clockwise' gene (gene 2); a length 2 schema is the 'start quadrant' gene (gene 1); and a length 3 schema is the 'number of line-segments' gene (gene 3). Figure 8-24 also shows that a 1 bit schema is not disturbed by the single point crossover operator and that 2 and 3 bit schema are affected by a crossover probability of 0.6 such that the maximum value for K_c is 0.96 and 0.92 respectively. Thus the crossover operator will not disrupt the development of these schema more than 4% and 8% respectively. The maximum disturbance, for crossover probabilities of 0.8 is 0.89, i.e. an 11% decrease in the expected value for the 3 bit schema (gene 3). Even a crossover probability of 1.0 only reduces the expected number of 3-bit schema by 13%. The other two genes in the chromosome, used in Chapter 5, are both 5-bit schema. These two genes will be disturbed more than the 1-bit to 3-bit schema but the 'start contour search' gene (gene 4) can take a range of start positions, as a percentage of the complete contour, and still be able to find the appropriate start point on the contour. Hence the disruption due to the crossover operator can change this gene (gene 4) a large amount without inhibiting its capability to find the relevant start search point on the contour. The 'line-segment length threshold' gene (gene 5) normally evolves a threshold < 4, hence the relevant information in the gene only occupies two bits and is, therefore, not disturbed significantly by the high crossover probability.

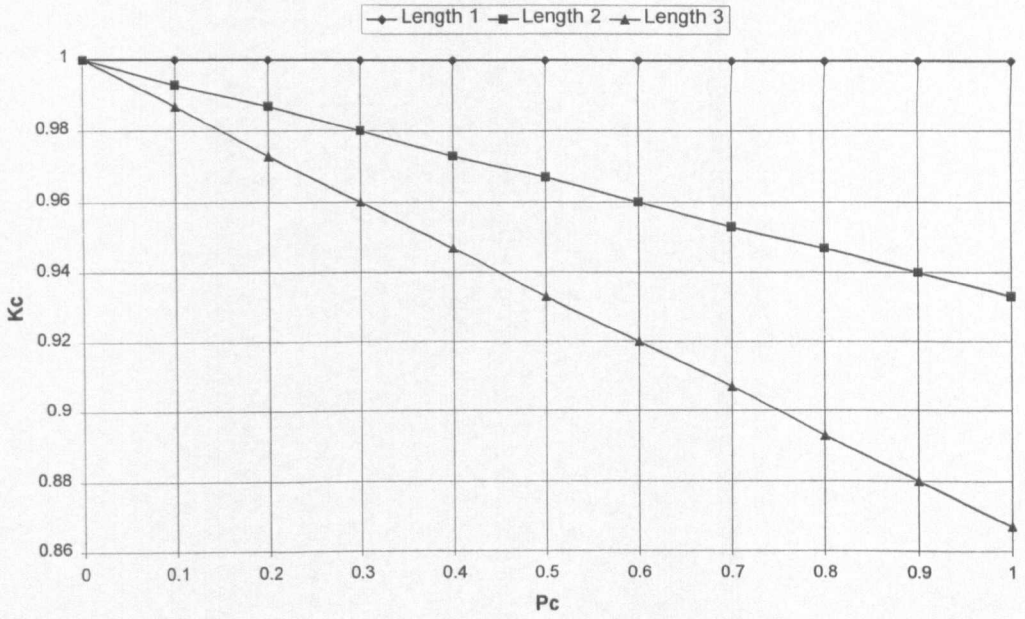

Figure 8-24: Kc values for Crossover Probability between 0.0 and 1.0

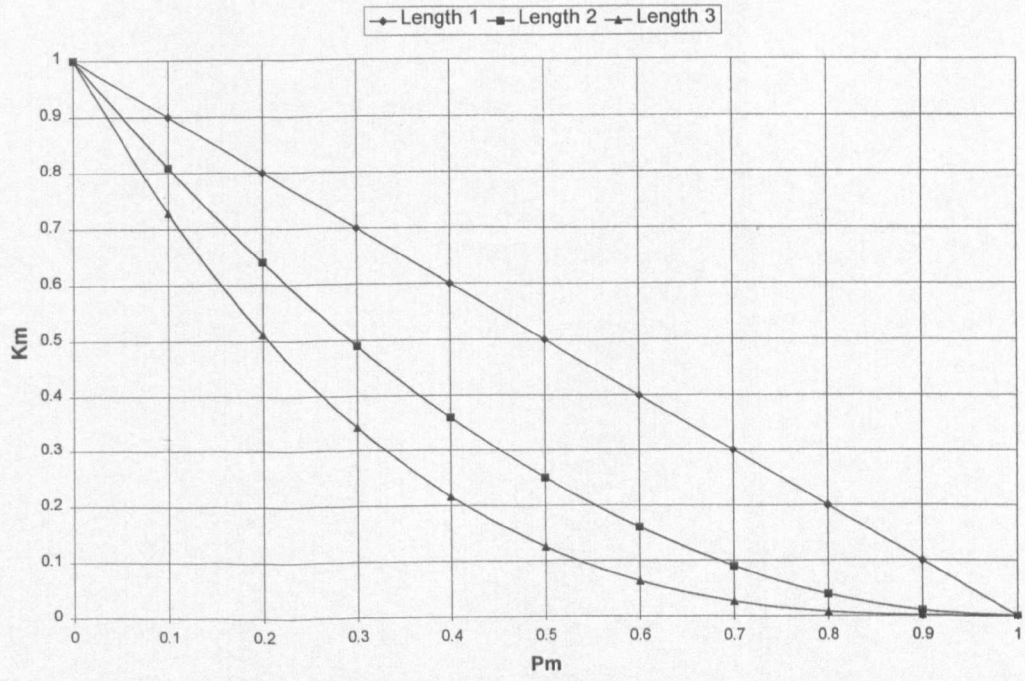

Figure 8-25: Km Values for Mutation Probability between 0.0 and 1.0

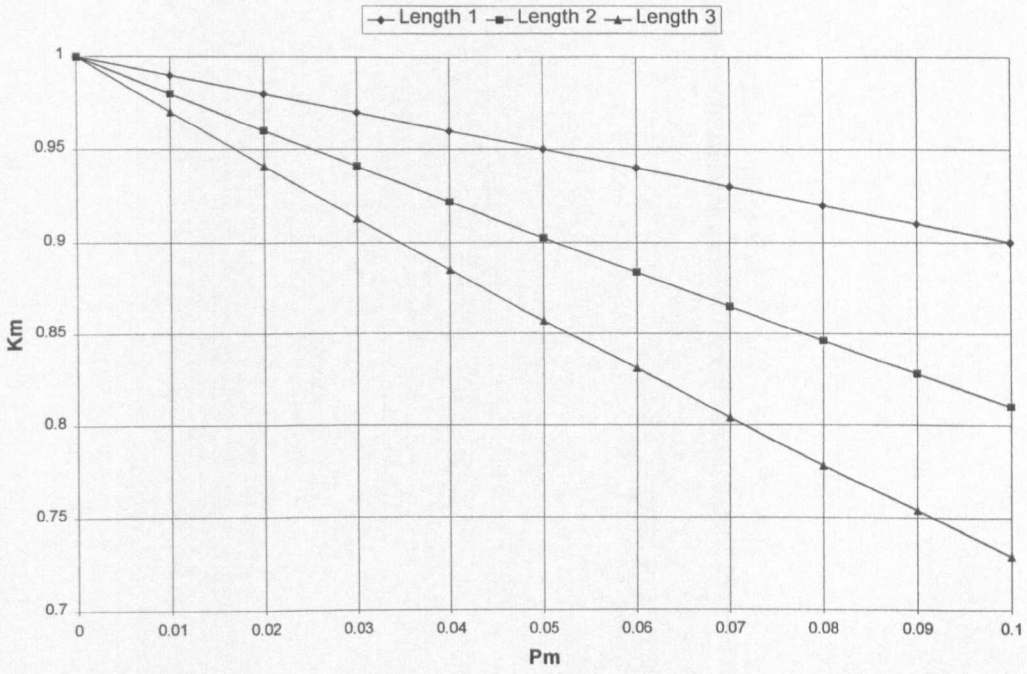

Figure 8-26: Km Values for Mutation Probability between 0.0 and 0.1

Similarly Figure 8-25 shows the effect of the various mutation probabilities on the expected values for the different length schema, i.e. 1, 2 and 3 bit schema respectively. From this set of curves it can be seen that for a mutation probability of 0.2, the disturbance (measured by K_m) to a 3-bit schema (gene 3) is of the order 0.5, i.e. only half these schema will be expected to survive into the next generation. Thus the diversity in the population is expected to be high. A value of 0.5 for K_m is equivalent to an almost random changing of the chromosome by the mutation operator. Test 12 to Test 16, Figure 8-1, show that mutation probabilities between 0.4 and 0.8 give diversity values similar to a random genetic algorithm. The K_m values for all three sizes of schema have values < 0.2 as the mutation probability varies over this range. These curves indicate why the mutation probability should be < 0.2 to allow the reproduction and crossover operators to 'explore' the solution space in an appropriate manner. The mutation probabilities should be sufficiently high to improve the diversity in the population without changing the search into a random process.

Figure 8-26 shows the variation in the value of K_m for mutation probabilities between 0.0 and 0.1 for each of the three schema sizes. These curves indicate that, for the usual values of mutation

probability for a 'standard' genetic algorithm, i.e. < 0.01, the disturbance to the different length schema is < 5%. Mutation probabilities in this range of values will allow the standard genetic algorithm to perform the crossover and reproduction operators in order to evolve the single 'best' solution. The genetic algorithm developed in Chapter 5 is required to find a range of less fit solutions in order to improve the recognition capabilities of the evolved chromosomes. Hence the mutation probabilities have to be set higher to provide the necessary diversity in the population.

8.4 Chromosome Fitness Analysis

Davidor (1991, page 33) discussed the 'Epistasis' of a chromosome, where epistasis designates in biology the effect of one gene on the expression of another gene, so that if such an effect exists the affected gene is said to be epistatic to the effecting gene. Davidor noted that "The literature on genetic algorithms emphasises that genetic algorithms do not 'see' the problem domain because the latter is obstructed by the representation. Accordingly, the question: 'Which problem domains are suitable for a genetic algorithm search?' should be replaced with the question: 'Does the representation (of a problem) promote an efficient genetic algorithm?' Only certain degrees of nonlinearity enable a genetic algorithm search to exhibit a relative efficiency, yet others will diminish this efficiency."

Davidor identified the effect that epistasis has on the relative efficiency of a genetic algorithm as follows:

1. A representation with small epistasis means that co-adaptation is not prominent and, therefore, a hill climbing algorithm is more likely to be more efficient.
2. A representation with high epistasis implies that co-adaptation is very strong and therefore, the efficiency of a genetic algorithm will decrease significantly.
3. A representation with mild epistasis is suitable for a genetic algorithm.

Davidor also noted that, "It is now possible to appreciate that there are additional factors which

affect the expected performance of a genetic algorithm besides the nature of the building blocks. If the epistasis can be measured for a given representation, then it will offer an important yardstick of its suitability to a genetic algorithm."

Davidor (1991, page 120) also discussed a measure of epistasis based on the fitness of each *bit* in the chromosome. A version of Davidor's method for measuring the epistatic effect was implemented as part of the research work using the chromosome of the genetic algorithm developed in Chapter 5. The results using Davidor's method did not appear to give a clear indication of the interaction of the *bits* within this particular chromosome.

The following method of analysing the interaction between the *genes* in the chromosome was investigated in the research work, with a view to indicating whether the genetic algorithm was suitable for application to the shape recognition problem. The complete range of the chromosome values was analysed by modifying the standard genetic algorithm of Chapter 5 to cycle through all the 65536 combinations of the sixteen-bit chromosome values. The fitness distribution for the digit two is shown in Figure 8-27 and Figure 8-28, where Figure 8-28 shows a logarithmic scale to highlight the fittest solutions. The fitness distribution for the digit four is likewise shown inFigure 8-29 and Figure 8-30. Note that the fitness distributions are similar and that, for both digits, about 0.4% of the chromosomes have a fitness > 0.6 and that approximately 1.0% of the chromosomes have a fitness > 0.5.

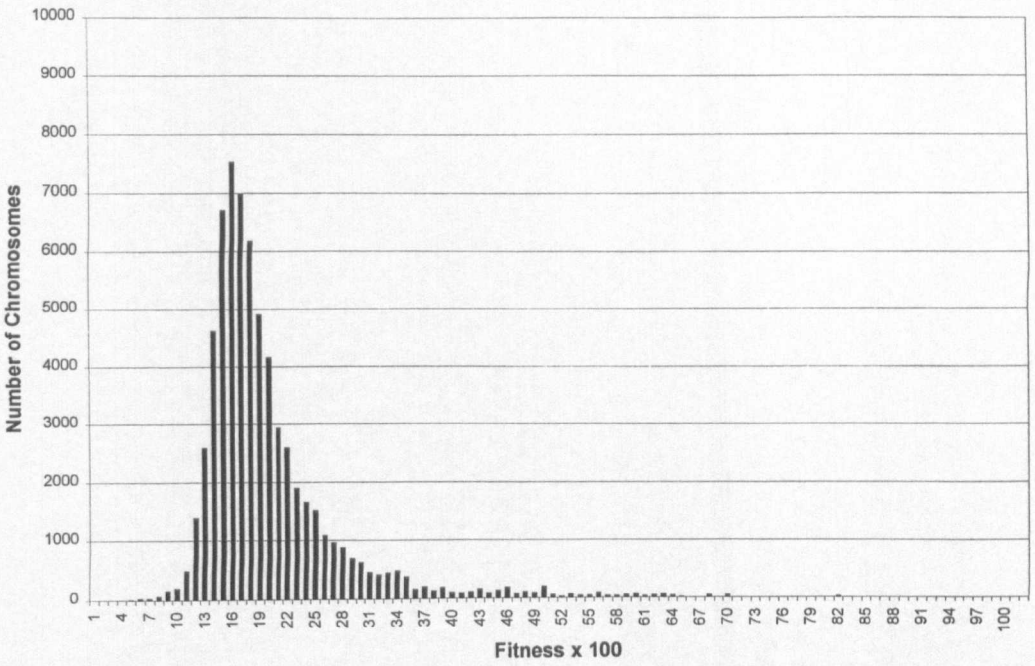

Figure 8-27: Digit Two : Fitness Distribution

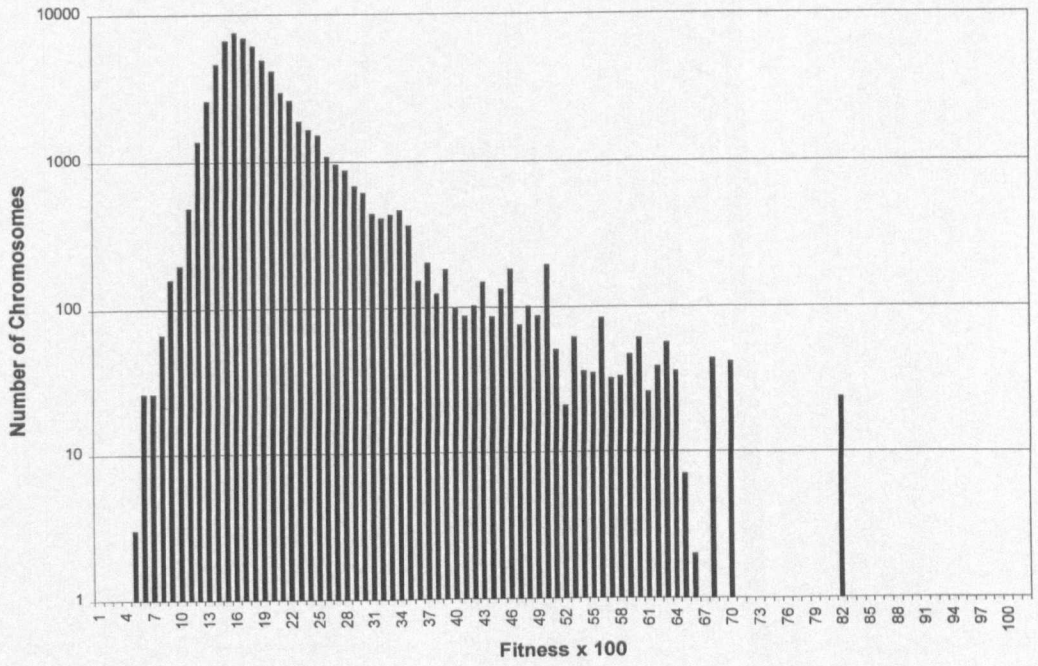

Figure 8-28: Digit Two : Logarithmic Fitness Distribution

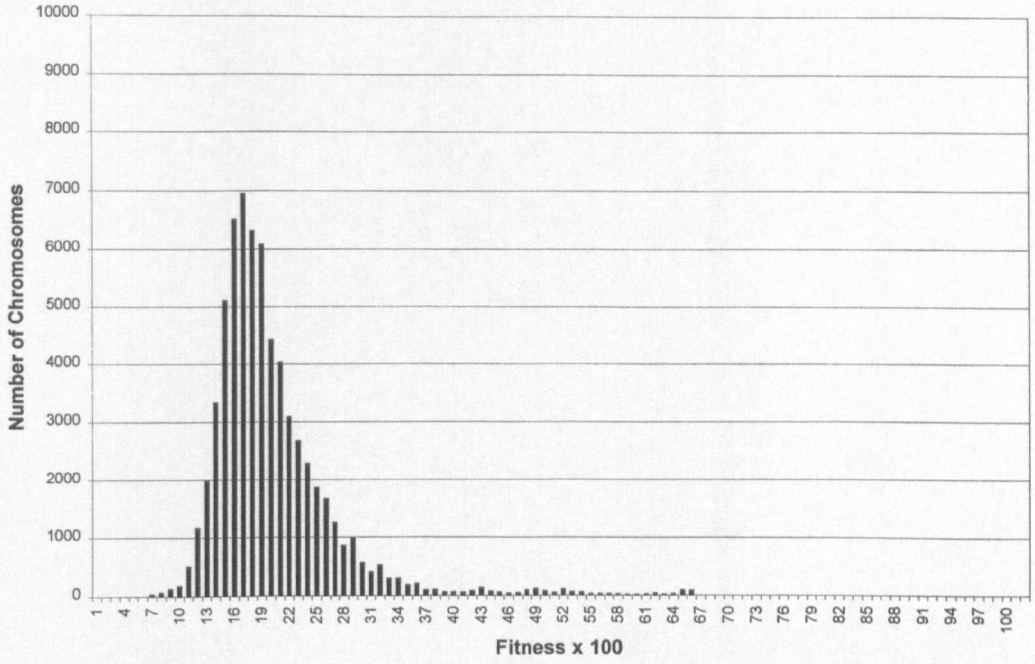

Figure 8-29: Digit Four : Fitness Distribution

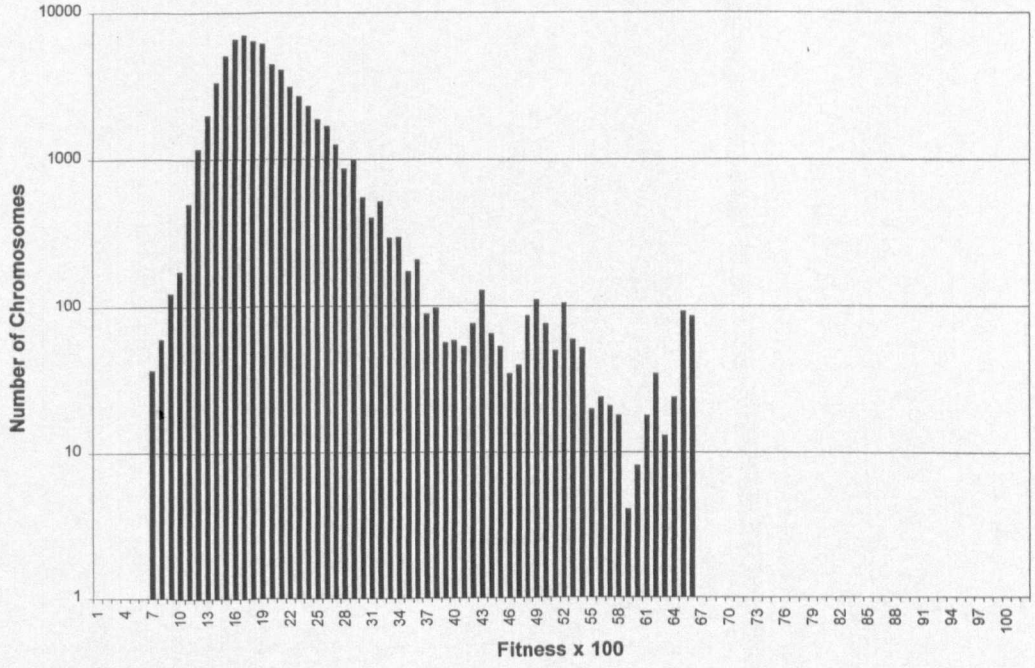

Figure 8-30: Digit Four : Logarithmic Fitness Distribution

The number of different solutions (using the observation genes 1, 2 and 3) with a fitness > 0.5 is shown in Figure 8-31 (digit 2) and Figure 8-32 (digit four). Column 2 shows the value of the 'observation genes' in the chromosome and column 3 identifies the individual values for genes 1, 2 and 3. Column 4 indicates the number of each gene combination, while columns 5 to 7 show the

minimum, average and maximum values of the fitness respectively. The chromosome 0-0-2 is identified as the best solution for both the digit two and the digit four and each of these digits has approximately the same number of different solutions. Apart from a fitness of 0.81, for the digit two, the fitness for the digits two and four is in the range 0.5 to 0.65. Thus each training set has strong internal similarities, i.e. high fitness values, when observed over different parts of the respective contours.

fx4.dif : fx4.pas ... SAT 16-MAY-1998 ::: 09:51:57
WED 06-MAY-1998 --- 19:38:07
d:\two.epi : 0.500

Number	Observation Genes Code	Chromosome	Count	Minimum Fitness	Mean Fitness	Maximum Fitness
1	0	0 0 2	212	0.514	0.638	0.809
2	3	0 0 5	37	0.511	0.605	0.635
3	4	0 0 6	23	0.513	0.595	0.630
4	9	0 1 3	5	0.505	0.537	0.559
5	10	0 1 4	30	0.505	0.557	0.583
6	11	0 1 5	27	0.515	0.565	0.591
7	12	0 1 6	27	0.513	0.567	0.592
8	16	1 0 2	80	0.513	0.577	0.617
9	24	1 1 2	45	0.578	0.612	0.624
10	32	2 0 2	25	0.508	0.527	0.562
11	48	3 0 2	121	0.514	0.561	0.642
12	50	3 0 4	3	0.554	0.568	0.586
13	53	3 0 7	3	0.514	0.517	0.518
14	54	3 0 8	8	0.504	0.518	0.530
15	55	3 0 8	8	0.504	0.518	0.530
16	56	3 1 2	27	0.536	0.581	0.617
17	57	3 1 3	12	0.577	0.599	0.617
Total Number of Chromosomes:			693			

Figure 8-31: Digit Two : Different Chromosomes with Maximum Fitness > 0.5

Number	Observation Genes Code	Chromosome	Count	Minimum Fitness	Mean Fitness	Maximum Fitness
fx4.dif : fx4.pas ... SAT 16-MAY-1998 ::: 09:55:26						
MON 04-MAY-1998 --- 19:09:12						
d:\four.epi : 0.500						
1	0	0 0 2	271	0.507	0.626	0.650
2	1	0 0 3	21	0.610	0.610	0.610
3	4	0 0 6	52	0.507	0.516	0.528
4	5	0 0 7	13	0.522	0.522	0.522
5	16	1 0 2	66	0.504	0.537	0.606
6	25	1 1 3	14	0.517	0.524	0.546
7	26	1 1 4	6	0.525	0.532	0.547
8	32	2 0 2	51	0.506	0.538	0.594
9	33	2 0 3	14	0.501	0.507	0.509
10	34	2 0 4	20	0.502	0.514	0.529
11	35	2 0 5	1	0.525	0.525	0.525
12	36	2 0 6	4	0.505	0.505	0.505
13	37	2 0 7	5	0.512	0.513	0.515
14	40	2 1 2	6	0.570	0.571	0.572
15	48	3 0 2	50	0.504	0.539	0.563
Total Number of Chromosomes:			594			

Figure 8-32: Digit Four : Different Chromosomes with Maximum Fitness > 0.5

The training sets for the digits two and four are shown in Figure 8-33 and Figure 8-34 respectively, together with their corresponding chromosome maps (Figure 8-35 and Figure 8-36). The chromosome map is similar to those shown in Chapter 5 and indicates the distribution of the chromosomes, above a specified fitness threshold (0.5), using the numerical values of the observation genes 1, 2 and 3 vertically and the parameter genes 4 and 5 horizontally. These chromosome maps show the distribution of local maxima in the population and also identify the 'gaps' between the local maxima groups. Further research on the analysis of this type of map is suggested to indicate what measure of epistasis can be obtained from the distribution of local maxima.

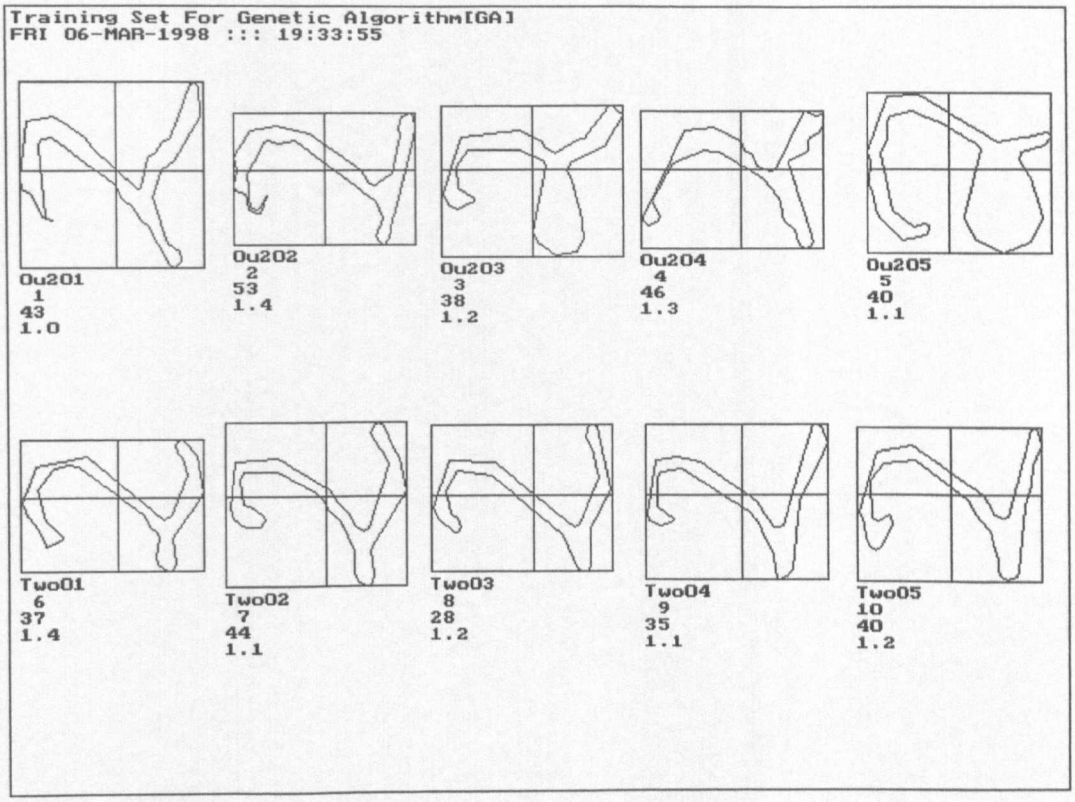

Figure 8-33: Digit Two : Training Set

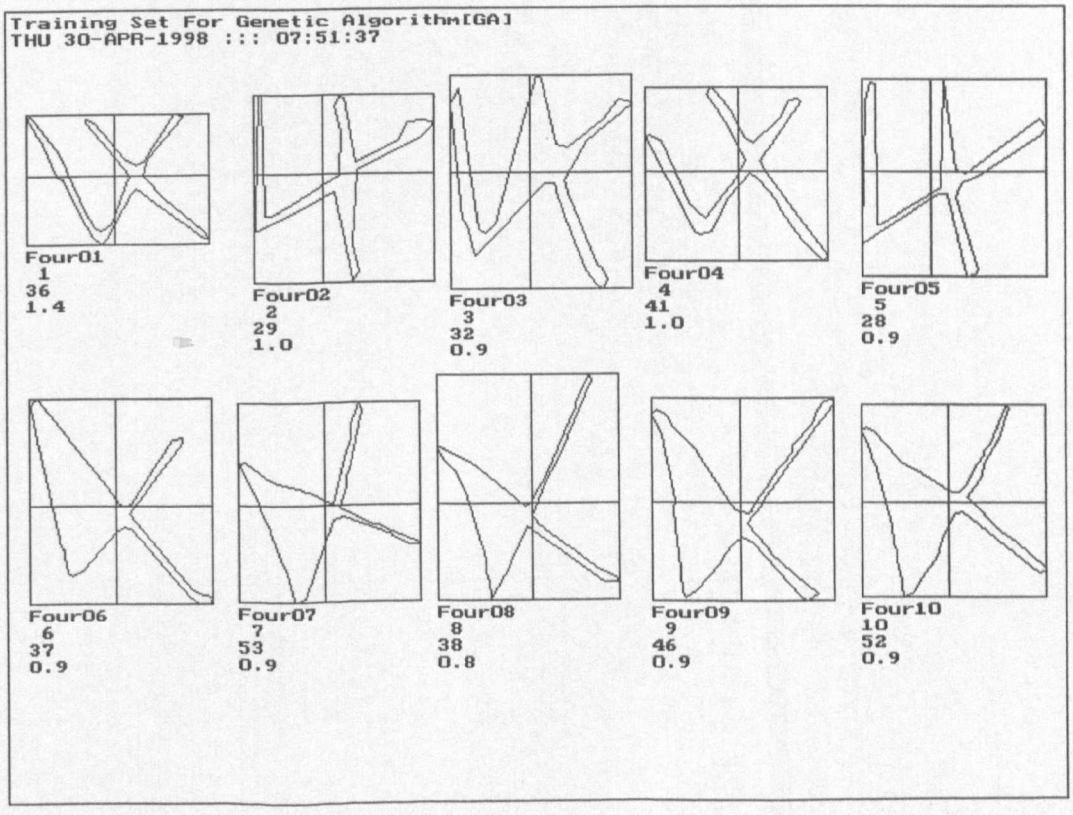

Figure 8-34: Digit Four : Training Set

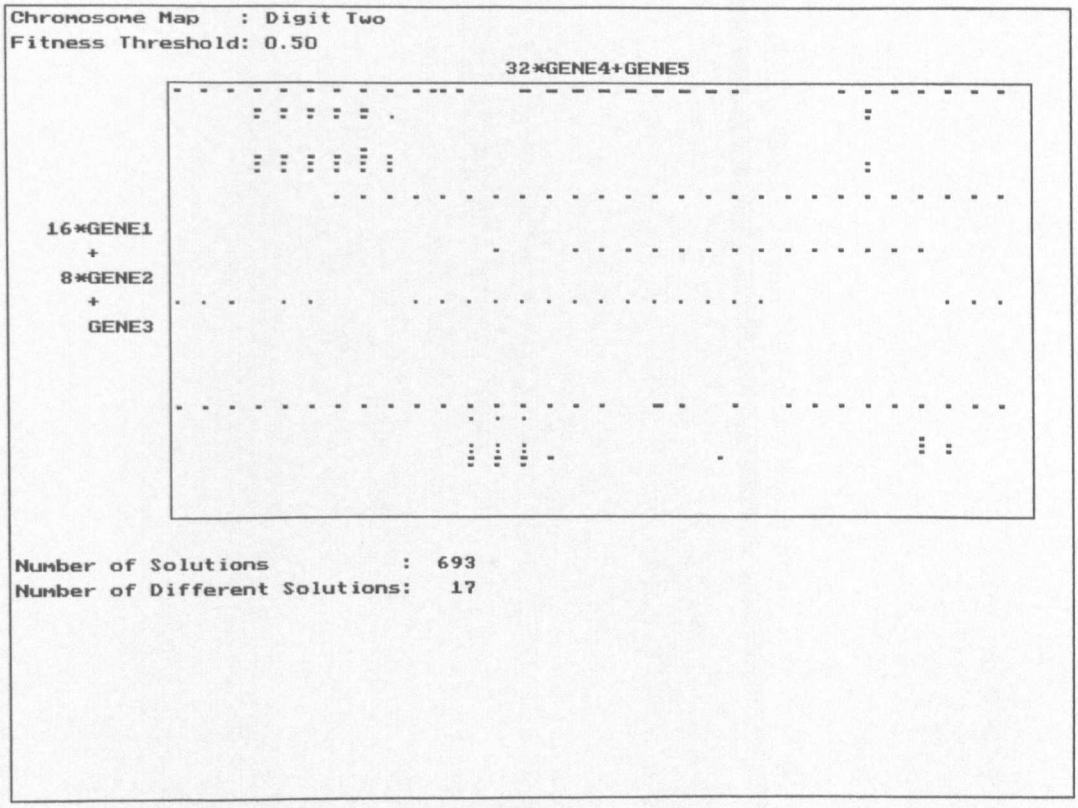

Figure 8-35: Digit Two : Chromosome Map

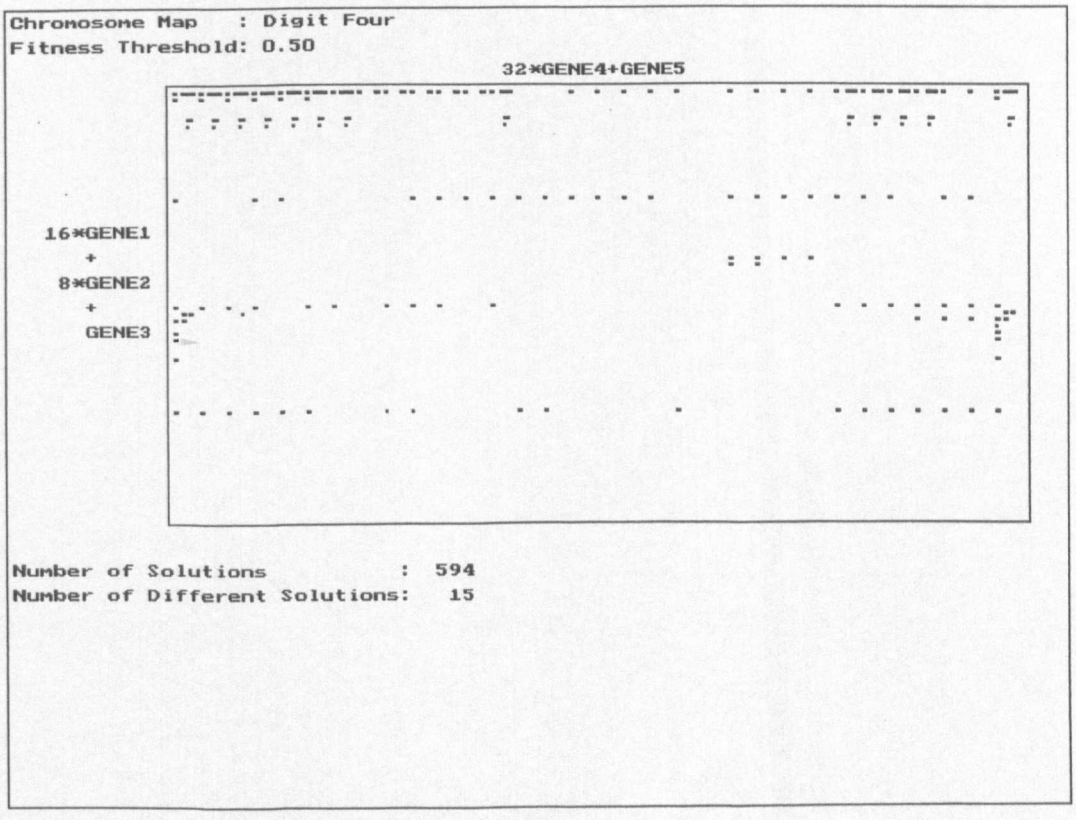

Figure 8-36: Digit Four : Chromosome Map

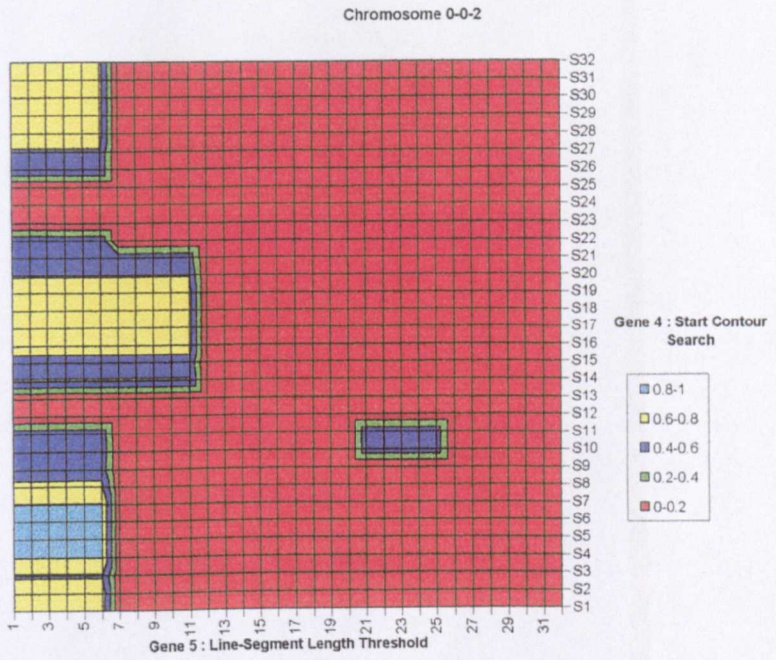

Figure 8-37: Digit Two : Gene 4 by Gene 5 Fitness Map : Fitness > 0.5

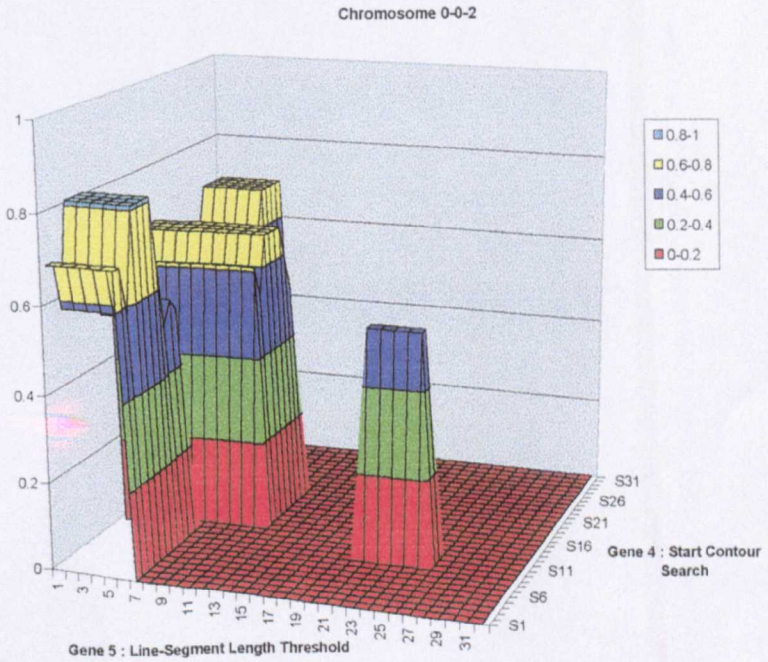

Figure 8-38: Digit Two : Gene 4 by Gene 5 Fitness Distribution : Fitness > 0.5

The effect of the various values of the individual parameter genes 4 and 5 on the fitness, for the digit two, of the combination 0-0-2 of the observation genes 1, 2 and 3 is shown in Figure 8-37 to Figure 8-40.

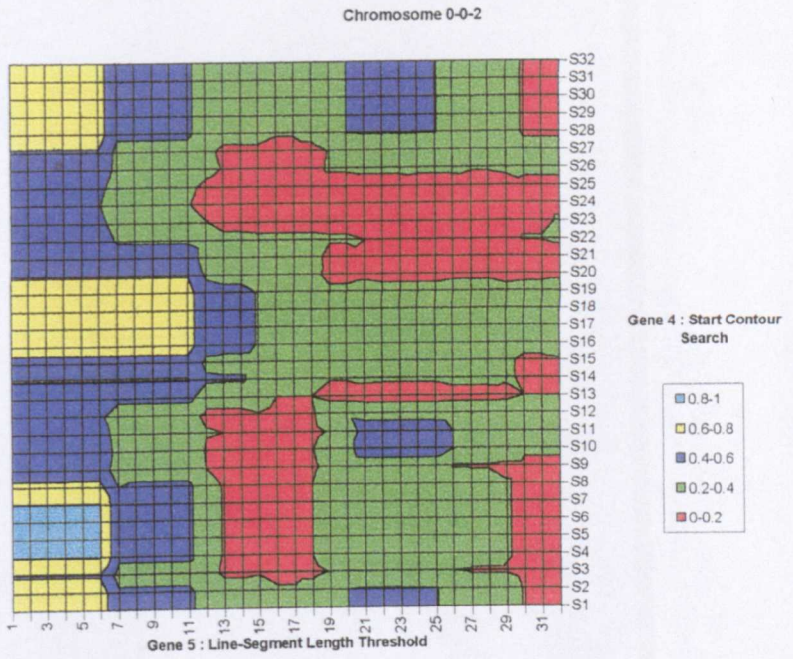

Figure 8-39: Digit Two : Gene 4 by Gene 5 Fitness Map : Fitness > 0.0

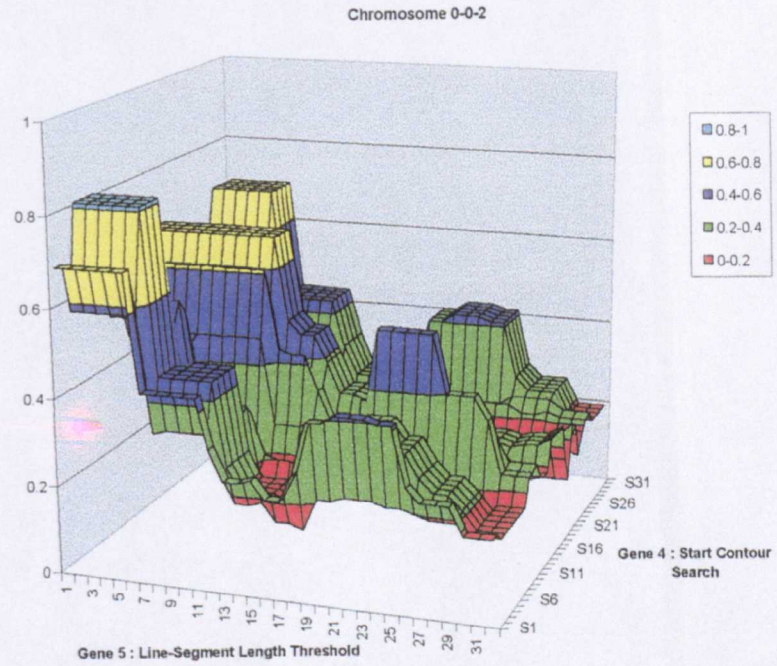

Figure 8-40: Digit Two : Gene 4 by Gene 5 Fitness Distribution : Fitness > 0.0

The variety of local maxima can be seen together with an indication of the parameter genes 4 and 5 values that generate low fitness for the observation genes 1, 2 and 3. Similar fitness distributions are shown in Figure 8-41 to Figure 8-44 for the digit four genes. Further research analysing this form of the fitness distribution would also be required to measure the epistasis of the gene interactions. This

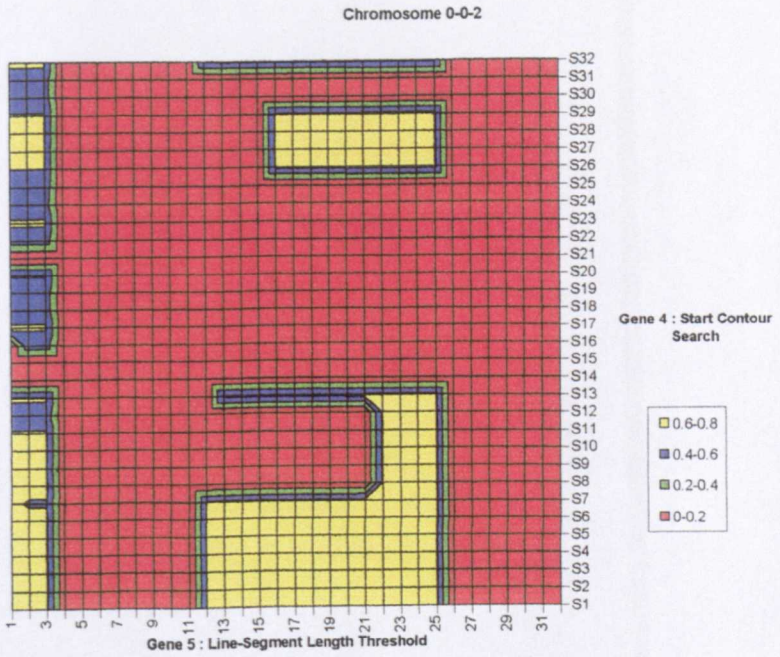

Figure 8-41: Digit Four : Gene 4 by Gene 5 Fitness Map : Fitness > 0.5

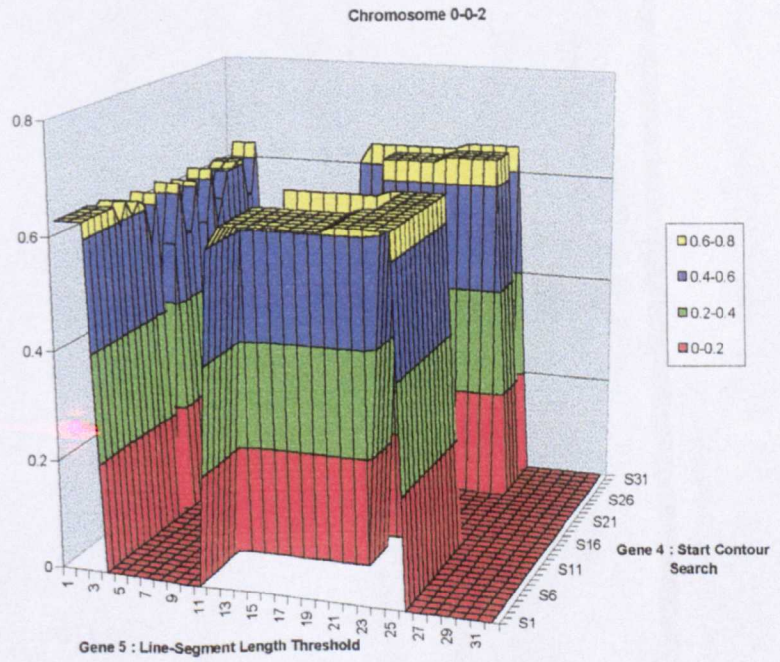

Figure 8-42: Digit Four : Gene 4 by Gene 5 Fitness Distribution : Fitness > 0.5

future research should include an analysis of the different contours in these figures in order to discover if this analysis can indicate how good the gene combinations are for contour shape recognition.

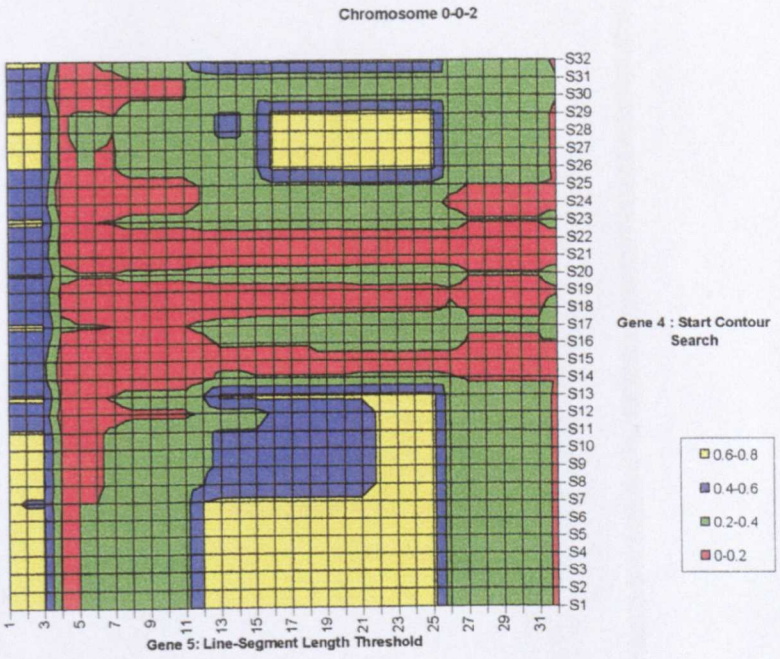

Figure 8-43: Digit Four : Gene 4 by Gene 5 Fitness Map : Fitness > 0.0

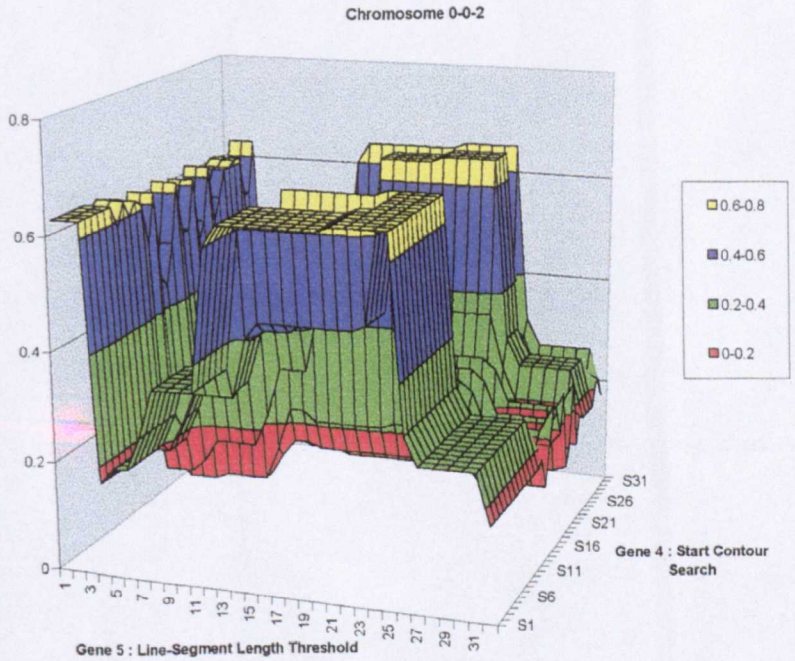

Figure 8-44: Digit Four : Gene 4 by Gene 5 Fitness Distribution : Fitness > 0.0

Chromosome histograms for the digits two and four are shown in Figure 8-45 and Figure 8-46, where the histogram bins are in units of 64. This form of representation shows another way of looking at the chromosome distribution and clearly indicates that the two distributions are not the

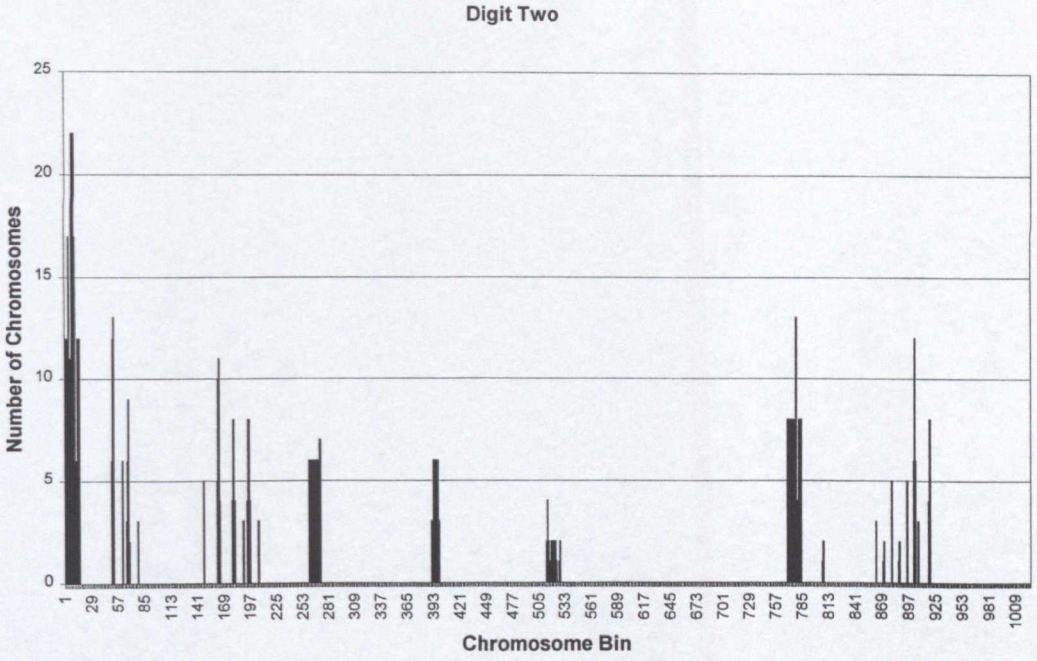

Figure 8-45: Digit Two : Chromosome Histogram : Fitness > 0.5

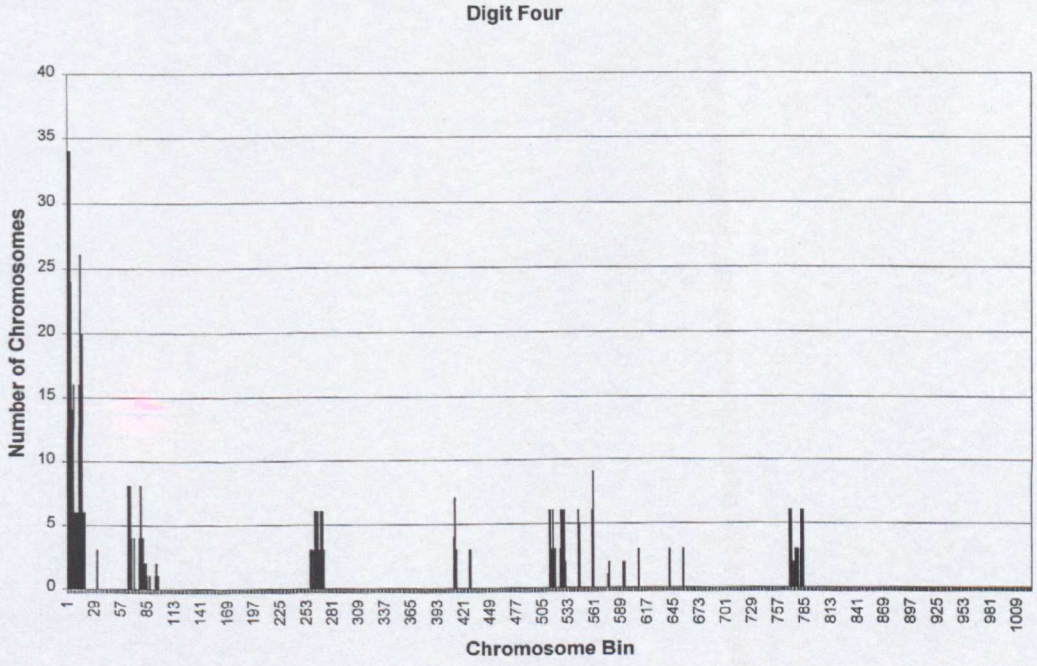

Figure 8-46: Digit Four : Chromosome Histogram : Fitness > 0.5

same. Further research is suggested to develop a method for analysing this form of distribution in order to supplement the shape features already supplied by the PIP line-segment vector information for the contour.

8.5 Summary and Conclusions

The research experiments reported in this chapter show that the roulette wheel selection process used by the standard genetic algorithm in Chapter 5 followed the theory of an exponential increase in the number of chromosomes that have above-average fitness. The constant of proportionality varies with time and approaches 1.0 as the population evolves to one solution (see also Figure 8-5). This variation in time is not generally mentioned in the literature but is discussed in Holland (1992).

A value for the crossover probability, of 0.6 to 0.8, provided sufficient 'exploration' of the solution space as shown by medium diversity values of the population for this range of probabilities (Figure 8-1, Tests 2 to 6). This range of crossover probabilities would enable the genetic algorithm to evolve a number of less fit solutions as well as potentially developing the 'best' solution. The use of the crossover operator by itself may be sufficient to develop the necessary number of different solutions required by the recognition process. A medium population average fitness also occurs with the use of the crossover operator by itself.

Mutation probabilities, in the range 0.4 to 0.8, gave population diversities similar to those of a random genetic algorithm (Figure 8-1, Test 12 to Test 16). Mutation probabilities < 0.4 reduced the diversity slightly and also reduced the disturbance to the actions of the selection and crossover operators. The use of the mutation operator does increase the diversity of the population and should therefore lead to the evolution of more variety in the solutions that are developed. More variation in the solutions would mean that more variety will be available to the recognition process and hence a better recognition capability is to be expected. These research experiments have shown that a mutation probability of the order of 0.2 was required to produce the necessary chromosome diversity for good recognition capabilities to be developed.

The research work described in this chapter has shown that the genetic algorithm developed in Chapter 5 behaved according to the schema theorem because the genes were of low order. As

mentioned above, the two five-bit parameter genes (genes 4 and 5) in the chromosome also behaved as if they are of low order. It could therefore be said that this chromosome was matched to the environment.

Mitchell (1996, page 158) discussed 'adapting the encoding' and stated that "Choosing a fixed encoding ahead of time presents a paradox to the potential genetic algorithm user. For any problem that is hard enough that one would want to use a genetic algorithm, one doesn't know enough about the problem ahead of time to come up with the best encoding for the genetic algorithm." Mitchell (1996) also noted that "... the ordering of bits is arbitrary, and may possibly impede the genetic algorithm from finding better solutions quickly. To find high-fitness rules, many bits spread throughout the chromosome have to be co-adapted. If these bits are close together on the chromosome, so that they are less likely to be separated under the crossover operator, then the performance of the genetic algorithm would presumably be improved."

Mitchell (1996, page 159) pointed out that Holland discussed proposals for adapting the encoding of the chromosome. Holland (1992, page 106) proposed an 'inversion' operator, which is a reordering operator inspired by a similar operator in biological genetics. Problems occur with this 'inversion' operator when crossover produces a chromosome that does not have the full complement of genes. One possible solution is usually to choose a second parent, which has the same ordering as the first parent. This problem with the 'inversion' operator limits the selection of parents for reproduction and may reduce the ability of the genetic algorithm to find the necessary solutions.

As a result of the research work described in this chapter, it is suggested that a possible method for adapting the encoding for the contour observation chromosome (Chapter 5) without using an inversion operator, would be to have a variety of populations with genes of different lengths in the chromosome. The best solutions from the different populations could be combined to provide an improved recognition capability. Different length genes could be developed which would still be of

low order but may have an advantage when observing the contour for shape recognition. The 'start quadrant' gene (gene 1) could be expanded to three or four bits so that a variety of 'start grid' configurations as shown in Figure 8-47 can be specified.

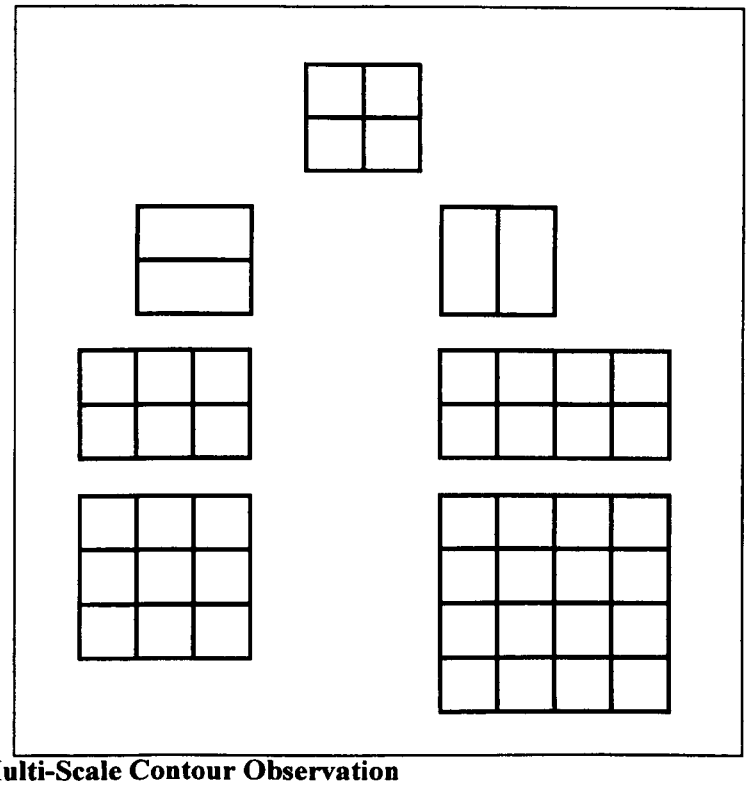

Figure 8-47: Multi-Scale Contour Observation

These grid patterns represent a simple form of multi-scale observation of the contour. The top pattern corresponds to the current genetic algorithm (Chapter 5) and the other grid patterns are represented by one, three and four-bit 'start grid' genes respectively. Each population would have a different sized 'start grid' gene so the chromosomes would be of different, **but fixed**, lengths in each population, hence avoiding the problem of variable length chromosomes that may occur with an inversion operator (see also similar comments from the genetic programming research in Chapter 3).

It is also proposed that a further modification within each population could involve increasing the length of the chromosome, so that multiple sets of the observation gene combinations that specify how to observe the shape contour are contained within the chromosome. The number of these sets could also be evolved and a chromosome would then define how to observe the contour at a number

(n) of different 'start grid' positions around the contour, with some of the 'start grid' genes potentially being a different size. The value of 'n' could then be optimised, for a particular set of training contours, by another genetic algorithm which would work on the different populations, with different size chromosomes, to find the best variations in the lengths and types of genes in the chromosomes. Observing a contour in a variety of ways should improve the recognition capability of the current genetic algorithm.

Rudolph (1994) analysed the convergence properties of the standard or canonical genetic algorithm with mutation, crossover and proportional reproduction applied to static optimisation problems. Proof is provided that, by means of homogeneous finite Markov chain analysis, the canonical genetic algorithm will never converge to the global optimum unless the 'best' solution in the population is maintained, i.e. *elitist selection* is used. Results are discussed with respect to the schema theorem. This paper noted that the schema theorem does not **imply** that the canonical or standard genetic algorithm **will converge** to the global optimum in static optimisation problems. It is suggested in this reference that "as long as the mutation operator is applied in the usual manner, the only chance for making the canonical genetic algorithm converge to the global optimum is to modify the selection operator."

The genetic algorithm developed in Chapter 5 in general did converge. The addition of elitist selection to the genetic algorithm in Chapter 5 would be a suitable topic for future research (see also Yao and Sethares, 1994). This research is necessary to find out if, with the addition of an elitist selection process, the genetic algorithm of Chapter 5 would *always* converge even when the mutation probability is set to values of the order of 0.2 so that a number of 'good' solutions would evolve.

Davidor, in Rawlins (1991, page 23) further discussed 'epistasis variance' and noted that there is a general consensus that the coding of a problem domain holds an important key to a successful

genetic algorithm application. His paper suggested a simple statistic (epistasis variance), which is a regression analysis predicting the function value from the individual bits in the chromosome, as a means to measure the amount of non-linearity in a representation.

This paper also discussed genetic algorithm *Hardness* and identified three contributing elements to hardness:

1. The structure of the solution space.
2. The representation of the solution space.
3. The sampling error as a result of finite and often small populations.

and concluded that, "These three elements are not necessarily linked, and furthermore, the effect of each of them on genetic algorithm hardness is not fixed."

De Jong (1997) developed an empirical methodology for studying the behaviour of evolutionary algorithms based on problem generators. Three generators were described that could be used to study the effects of epistasis on the performance of the standard genetic algorithm. Multimodal problem generators that allow an increase in epistasis without significantly reducing fitness variance were also discussed. Initial results were presented that support the notion that the relative advantage of crossover over mutation is reduced as epistasis increases. The continued effectiveness of crossover at higher levels of epistasis was also reported.

The experiments on chromosome epistasis performed during the research work described in this chapter suggest that future epistasis research should be conducted along the lines suggested by Davidor and De Jong above, but identifying the interaction between the individual *genes* rather than the *bits* in the chromosome. The research methods developed in this chapter, to analyse the behaviour of the genes in a chromosome, should therefore be further developed.

9. Summary and Conclusions

9.1 Summary

The research work described in this thesis has investigated the use of the genetic algorithm for shape recognition in three main areas of application:

1. Evolving chromosomes that can detect the position of bright and edge pixels in an image.

2. Evolving chromosomes that can trace a contour in an image.

3. Evolving chromosomes that can specify how to observe parts of a contour.

These investigations concentrated the research work on the task of obtaining shape boundary contour features in a suitable form for input unto a genetic algorithm. The research then analysed the behaviour of the genetic algorithm for various values of the crossover and mutation probabilities. It was identified that a number of 'good' solutions **must** be evolved for recognition to be possible. A new recognition technique was developed that used these evolved solutions with the same fitness function as the genetic algorithm. The characteristics of fitness functions were analysed with special reference to avoid calculating incorrect local maxima. Thegenetic algorithm, developed to evolve a number of 'good' solutions, was investigated by the schema theory and epistatic analysis. It was shown to behave as expected of a standard or canonical genetic algorithm.

The research described in this thesis is summarised below and the various conclusions drawn from the experimental results are also provided. The research work has investigated the difficult task of shape recognition and the techniques developed during the research have enabled a number of suggestions for further research to be made.

9.1.1 Genetic Algorithm Input

Using the genes in a chromosome to control the motion of a cell was useful for cueing areas of high

intensity in an image. Fitness functions identifying edges in an image were been investigated. This type of chromosome encoding has been able to track the brightness and edge characteristics of a moving object. A fitness sharing function was used to inhibit the cells from migrating to a single position in the image. Experiments were performed, using genetic programming principles to evolve chromosomes that could trace a shape contour and store an encoding of the contour in the genes of the chromosome. The encoding contained in the chromosomes was also been used to attempt a reconstruction of the contour by decoding the information in the chromosome genes. The encoding process was successful but the decoding method, which is similar to a Travelling Salesperson Problem (TSP), appeared to be unsuccessful.

Methods have been investigated and developed for obtaining information about a contour that was in a form suitable for use by a standard or canonical genetic algorithm. In order to compare contours it is essential that they are normalised, so that any comparison can be made independently of translation, scale, rotation and starting point. Normalisation for skew was not been considered in this paper. In order to improve the recognition capabilities of Fourier descriptors and moment descriptors, it is necessary to obtain information about the Perceptually Important Points (PIPs) or dominant points on a contour. A simple method was developed that finds the predominant curvature changes on a contour. These points are the positions of high curvature along a contour. A multi-scale analysis of the contour was discussed that varies the spatial bandwidth of the contour, using for example a Gaussian shaped low pass filter in the Fourier domain, and varies the length of the sample points that are involved in the curvature calculations.

The design of a format for the contour information that is in a form suitable for input to a recognition process or genetic algorithm has proved to be difficult. A large proportion of the referenced papers on the application of the genetic algorithm to image processing problems apply the genetic algorithm to optimise the various parameters associated with each particular image processing recognition problem. This form of optimisation did not seem suitable for the PIP

information of a shape contour. Finalising the design of the contour information effectively fixed the structure and behaviour of the chromosomes in a genetic algorithm and also defined how the fitness of a chromosome was calculated. The process of matching the environmental information about a contour to the chromosomes in a genetic algorithm was not trivial, and, in practice, this process occupied a considerable length of time during the research.

9.1.2 Genetic Algorithm Behaviour

A set of normalised line-segment vectors of the contour PIP was designed as input to a genetic algorithm. The chromosome was structured to make use of this type of input and was able to develop a recognition mechanism that could complement that provided by the Fourier descriptors and moment descriptors.

A standard genetic algorithm was designed and investigated using binary and integer chromosomes. The chromosome was structured such that the various genes specified how the line-segment vectors were to be 'looked at' or 'observed' as groups. Typically a population of fifty chromosomes, evolving for thirty generations, was used for most of the tests. The various ways to observe a contour that have high fitness were evolved using a training set of contours. The genetic algorithm can also be run using a 'mismatch' fitness function i.e. (1.0 – measured fitness) so that groups of line-segment vectors can be evolved that indicate which parts of the contours in the training set do not match. This extra information will be useful in adding to the recognition capability of the genetic algorithm solutions.

Various settings for the mutation and crossover probabilities were investigated. For low values of mutation probability, < 0.2, the 'best' solution (fitness = 0.81, for the digit two) was developed with the average fitness of the population increasing as the evolution proceeded. Tests using low values of mutation probability showed that the genetic algorithm performs as expected, and exhibited the usual characteristics of the standard genetic algorithm as reported in the literature. Binary and

integer chromosomes appeared to behave in a similar manner. In general, a selection of solutions was required for the contour recognition process, so that the variation contained in the training set contours could be adequately covered by the different ways to observe each contour, as specified by each chromosome. In order to search the solution space for a selection of solutions, the mutation probability has to be increased to a value in the region of 0.2.

A chromosome diversity measure was designed that measured the standard deviation of the genes in the chromosome that specify how to observe the contour (observation genes). This diversity measure was compared to one, reported in the literature (Lis, 1995), using fitness dispersion. Both methods indicated the level of diversity that is required to adjust the value of the mutation probability in order to evolve a variety of different good solutions rather than just the best solution.

9.1.3 Contour Recognition

Each of the chromosomes evolved by the genetic algorithm has associated with it the mean and standard deviation data for each of the line-segment vectors appropriate for the training set of example contours. This set of mean and standard deviation data for each line-segment is referred to as the 'set of features' for that part of the contour examined by the instructions in the chromosome. Shape recognition requires some form of classification method that is able to partition the various distinguishing features of a shape, and its contour, into groups or classes such that the overlap between the classes is small. The use of the genes in the evolved chromosomes enables a simple recognition process to be achieved that requires two parameters, a fitness threshold (Rthreshold) and a recognition threshold (R).

Correct recognition can be achieved by a choice of chromosome, fitness threshold and recognition threshold. The values for these two thresholds can vary for each chromosome and is thus a flexible feature of the simple recognition method. The fittest chromosomes are not necessarily the best for correct recognition, and when used in combination the chromosomes can provide a recognition

capability with few false alarms. The genetic algorithm also provides a mismatch capability, which can be used as extra evidence for contour recognition. The false alarms can be reduced to zero by a choice of chromosome, fitness threshold and recognition threshold.

9.1.4 Fitness Calculation

Certain combinations of line-segment vectors show high fitness values, but the contour shapes do not appear similar to the human observer. Likewise certain combinations of line-segment vectors have low fitness and the contours appear to be similar. Provisional analysis suggests that these effects are due to the search start point in the input contour data being too sensitive to noise on the measured contour line-segment vectors and the calculation of the two dimensional fitness value suffering from a non-linearity and/or an asymmetry. The non-linearity and/or asymmetry in the fitness function output over different ranges of the fitness value has to be considered when using a ranking method for the selection process. The range and slope of the ranking function will have to be adjusted according to the *fitness values* and the *parameter values* used in the fitness calculation.

Fitness calculations for a variety of shapes have been analysed to discover if any local maxima or minima occur in the fitness values. Local maxima in the fitness function caused the genetic algorithm to incorrectly bias the selection process towards the local maxima solutions at the expense of the correct solutions. Certain fitness functions can generate local minima in the fitness values, thus biasing the selection process in the genetic algorithm away from the chromosomes that will contribute to the 'best' solutions. This type of defect in the fitness calculation is not as serious as local maxima in the fitness function, but the performance of the genetic algorithm can be reduced.

Kinnear (1994, page 9, section 1.4.2) discussed the *exceptional* importance of the fitness function. "You simply cannot take too much care in crafting your fitness function. Any and all boundary conditions, in your fitness function, will be exploited ruthlessly by the individuals in your population. Outright, though subtle, bugs in your fitness calculations will almost certainly be

recognised by the evolutionary process, and the only way in which this can be determined is to examine the individuals that evolve." Kinnear (1994), in the same section of this reference, also highlighted the following:

1. A central aspect of fitness functions is the concept of *partial credit.* A fitness function needs to score an individual on how well it performs on the problem to be solved. "However it needs to do more than this, it needs to score individuals in such a way that they can be compared and a more successful individual can be distinguished from a less successful individual."

2. Any fitness function must be able to evaluate partial solutions to a problem and produce a score that distinguishes a more complete solution from a less complete solution. The favouring of one partial solution over another defines the direction that the evolutionary processes encourage individuals in the population to move.

3. In addition to partial credit, there are other factors to consider when designing a fitness function. "One of these factors is how to aggregate the results of multiple fitness tests, in those cases where the overall fitness of an individual is determined by its performance during several separate fitness tests."

This last point is relevant to the design of the fitness function for the genetic algorithm described in Chapter 5.

9.1.5 Schema and Fitness Analysis

The behaviour of schema in the shape contour observation chromosome has been examined, as the crossover and mutation probabilities are varied in order to increase the diversity of the population. The analysis investigated the relative effect of crossover and mutation in discovering a variety of less fit solutions, rather than just evolving the 'best' solution. In order to reduce a possible data explosion, only the schema defined as 'observation genes' are monitored.

A possible method suggested for adapting the encoding for the contour observation chromosome, without using an inversion operator, is to have a variety of populations with genes of different lengths in the chromosome. The start quadrant gene could be expanded to three or four bits, so that a variety of grid patterns would be superimposed on the shape contour. These grid patterns represent a simple form of multi-scale observation of a contour. A further modification, within each population, would involve increasing the length of the chromosome so that multiple sets of observation gene combinations, which specify more than one way to observe the shape contour, could be contained in the chromosome.

The fitness distribution for all possible chromosome values was obtained for two sample shape contours (digits two and four) and recorded onto disc. The distribution of fitness for various combinations of genes was obtained in order to discover if the structure of the genes in the chromosome was suitable for an efficient genetic algorithm application. A discussion is provided that suggests that the 'epistasis' of a chromosome can be obtained from a measure of the interaction of the genes in the chromosome, rather than from any interaction between the 'bits' in the chromosome representation (see also Davidor, 1991).

9.2 Conclusions

9.2.1 Genetic Algorithm Input

Controlling the motion of a chromosome using a fitness function that gives strong responses to edges and bright areas in an image can be used with advantage to cue areas of interest in an image or in a sequence of images. The values of the genes in the chromosome indicate the direction of motion of the chromosome, but do not provide any form of description of the cued shapes. Hence this type of chromosome, with the genes specifying the direction of motion, would make a good tracker which could possibly be implemented as a fine-grained algorithm. A fitness sharing function is required to avoid premature convergence of the chromosomes to the same position in the image.

This sharing process is processor intensive, and may not be practical in a real-time system.

It is possible to train a chromosome to follow a contour, using a chromosome based on the genetic programming paradigm. The chromosomes evolve genes that control the actions taken by the chromosome as the contour is traced but, again, the values of these genes do not contain any information about the characteristics of the contour. The traced contour can be encoded using a list of the various genetic programming functions that are evolved specifically to trace the contour. Decoding this list of functions is equivalent to a TSP. Experimental evidence suggests that the solution to this problem is not achievable for realistic contours. The variable length property in a genetic programming chromosome is limited by the finite memory available on a processor, and the length of the chromosome has to be artificially curtailed.

The experiments performed using the genetic programming method confirm that the most appropriate encoding of a contour is in fact the traditional Freeman chain coding (see Freeman, 1961 and 1977). This type of contour encoding identifies a numerical direction for each point on a contour. Even with this type of coding, the problem of how to encode the contour and its features in a way that is suitable for input into a genetic algorithm still remains. It is considered that some form of encoding the high curvature parts (PIPs or dominant points) on the shape contour can be an appropriate method for providing an input to a genetic algorithm.

The co-ordinates of the closed shape contours must be normalised for start point, translation, size and rotation before any features such as PIP, Fourier descriptors or moment descriptors can be used for recognition purposes. Fourier domain filtering can perform this normalisation and can also smooth the closed contour before transforming back to the spatial domain. A simple method has been devised for identifying the PIP on a closed contour. This method is similar to, but slightly different from, other methods reported in the referenced literature. The region of support for the PIP sampling of the contour can be varied and smoothing operations, at multiple scales, are possible by

low pass filtering in the Fourier domain.

Designing a format for the PIP information as input for a genetic algorithm is a difficult process and, in practice, took a long time to finalise. The final format is in the form of line-segment vectors (length and direction) as identified by a quadrant grid placed over the closed contour. The line-segment vectors are obtained from a compression of the collected PIP information, by joining together neighbouring sequences of line-segment vectors that have the same direction. The start and finish quadrants for these compressed line-segment vectors also forms part of the input information. Variation in the PIP support region and the amount of spatial filtering applied enables a fine-to-coarse sampling of the closed contour possible. Individual line-segment vectors or groups of line-segment vectors are available for analysis, so partial occlusion may not be detrimental to the overall input into the genetic algorithm.

9.2.2 Genetic Algorithm Behaviour

A chromosome can be designed that specifies how to 'look at' or 'observe' a contour. The genes define in which quadrant to start looking and how many line-segment vectors to include in each contour feature. It has been shown that the crossover and mutation parameters of the genetic algorithm **must** be set such that a number of less fit solutions are evolved, because the 'best' solution may not provide good recognition of contour features. Chromosome diversity measures have been investigated, and are shown to be able to indicate the crossover and mutation parameter values that are necessary for multiple solutions to be evolved. The genetic algorithm uses a triangular shaped fitness function and calculating the fitness of a chromosome from the sum of the individual line-segment vector parameters appears to be suitable for this application. Problems with the two-dimensional fitness calculations have been identified and are noted in the next section.

The results for binary and integer chromosomes do not appear to be significantly different. Evolving fifty chromosomes for thirty generations is usually sufficient for developing solutions that can be

used for contour recognition. Chromosomes that can identify a mismatch between the contours in a training set can be evolved using a fitness function for selection that is (1.0 – measured fitness). This mismatch information can be used as extra evidence by a recognition process. The genetic algorithm behaved as a standard or canonical genetic algorithm for mutation probabilities < 0.2, i.e. evolved a single 'best' solution. A number of less fit solutions are evolved for mutation probabilities > 0.2. A number of less fit solutions are required to cover the variation of the contours in the training set and in some cases the 'best' solution is not suitable for use by the recognition process. A fitness dispersion measure and/or a chromosome diversity measure appear to indicate the domain of the genetic algorithm in which a number of solutions will be evolved. The possibility therefore exists that these measures can be used to adjust the crossover and mutation probabilities at each generation so that a number of solutions evolve.

9.2.3 Contour Recognition

The tests described in Chapter 6 show that it is possible to use evolved chromosomes that contain 'observation and parameter' genes and the triangular fitness function to provide a recognition capability. Correct recognition can be achieved by a choice of chromosome, fitness threshold (Rthreshold) and a suitable recognition score threshold (R) parameter. A second genetic algorithm, operating concurrently with the genetic algorithm that evolves the observation chromosomes, could be used to optimise these parameters. A parallel implementation of the two genetic algorithms could be developed with the possibility of unsupervised learning being achieved. The approach to recognition taken in Chapter 6 is different to the usual methods reported in the literature. Identification of the correct position to start searching any input list of contour line-segment vector information remains a practical problem.

9.2.4 Fitness Calculation

The design of a fitness function that, for contours which are more like each other (as confirmed by a human observer) correctly calculates a higher fitness value, also remains a problem. The fitness

calculation of a genetic algorithm should always be analysed for the presence of local maxima. Local fitness maxima will produce a convergence onto incorrect solutions. Local minima can also be present in the output of a fitness function. This defect is not as serious as a local maximum, but can reduce the performance of the genetic algorithm. Fitness calculations can also be asymmetric. This property must be taken into account if ranking of the fitness is being used prior to chromosome selection for reproduction.

9.2.5 Schema and Fitness Analysis

In order to reduce the data explosion, schema analysis can be performed on individual genes rather than on particular bit patterns in the chromosome. The evolution of low order genes, for mutation probabilities < 0.2, follows the schema theorem, and the number of the fittest solutions grows exponentially each generation. The 'growth constant' varies with time and is generally ≥ 1.0 for the solutions with the 'best' fitness. The genetic algorithm using crossover probabilities between 0.6 and 0.9 combined with mutation probabilities between 0.1 and 0.3 can evolve a number of chromosomes that are useful for a recognition process. This range of values for the crossover and mutation probabilities provides a high diversity in the population, a reasonably large number of unique (individual) solutions (20 to 30) and a variety of different (i.e. 'observation' genes 1, 2 and 3) solutions (10 to 15). Applying the schema theory to genes, rather than to individual bits in the chromosome, can possibly give a better understanding of the evolutionary processes as the genes are developed. The 'observation' genes (genes 1, 2 and 3) in the chromosome are of low order and evolve according to the schema theory.

Chromosomes of different lengths can be used to sample a contour with a variety of grid patterns. The gene specifying the start grid (i.e. gene 1) can be increased in length to vary the number of grid patterns available to the line-segment observation process. Increasing the length of the chromosome, by including more than one set of genes 1 to 5, will make it possible to look at various parts of a contour at the same time. A recognition capability may then be possible during partial occlusion of

the contour.

The effect of one gene on another (epistasis) and also the effect of, say, the 'parameter' genes (genes 4 and 5) on the 'observation' genes (genes 1, 2 and 3), can be displayed using chromosome maps which show the areas on the fitness landscape where local maxima may occur. A histogram of chromosome fitness can be obtained for all combinations of chromosome values. An analysis of this type of histogram may identify particular patterns in the histogram that can then be used as extra information by a recognition process. It is noted that the production of chromosome fitness maps and fitness histograms for the complete population of chromosome values is a processor intensive activity.

9.3 Further Research Recommendations

As a result of the research, investigations and experiments reported in this thesis the following topics are suggested for further enquiry and research:

1. The measurement of the parameters of multiple contours using active contours (see Pardo, 1997) and the affine transform (see Ip and Shen, 1998 and Pei and Lin, 1995). Inesta (1998) gave a good summary of the problems associated with dominant point analysis such as noise and quantisation effects. The automatic adjustment of algorithm parameters has been discussed and should be included in this investigation (see also Ip and Wong, 1997). Possible extra 'rules' should also be tried when collecting the contour PIP information. Slusek (1995) proposed a novel adaptation to the moment descriptors of a shape that ought to be studied for application as an extra descriptor for a shape contour. Linear filtering of the shape contour in the form of wavelet descriptors for multi-resolution recognition of contours (Wunsch and Laine, 1995) is suggested as an addition to, or an alternative for, the low-pass filtering of a shape contour in the Fourier domain. Non-linear filtering of the shape contour before the PIP information is input into the genetic algorithm should also be further investigated (Brockett and Maragos, 1994).

2. The method of analysis of the fitness function for a 2D contour that can show that the fitness calculated does not have any local maxima should be investigated. The use of the Euclidean distance as a measure of similarity between contour features and the summation of individual feature fitness values should be included in this analysis. The triangular fitness function (see Chapter 5) should be further analysed to find out the best value for the number of standard deviations of a contour feature that are required for correct calculation of the line-segment fitness value. The fitness function analysis should also be extended to include an investigation into fitness ranking that uses fitness *parameter* values as well as fitness *values*.

3. An investigation into the convergence of a genetic algorithm that has its mutation and crossover probabilities set so that a number of 'good' solutions are evolved should be conducted (see Rudolph, 1994 and Yao and Sethares, 1994). This analysis should attempt to answer the following question: Will the genetic algorithm find a number of 'good' solutions consistently for differently shaped contours every time that it runs? The appearance and disappearance of less fit solutions has been observed (Chapter 5). The effects of this phenomenon on the answer to this question should also be studied.

4. An exploration of the use of the extended chromosome using a number of populations, each with a different length chromosome, as discussed in Chapter 8, should be conducted. The extended chromosome will contain genes that define a variety of grid patterns over the contour and/or a number of ways to observe different parts of the contour at the same time. Analysis of the design of a chromosome diversity measure for the adaptive adjustment of the crossover and mutation probabilities should also be further investigated (Srinivas, 1994). The chromosome diversity measure should be analysed to find out if this measure correctly indicates the domain where the genetic algorithm does produce the necessary variety of solutions for good contour recognition to be achievable.

5. A dual genetic algorithm environment should be implemented, where one genetic algorithm

evolves a number of 'good' solutions based on a training set of sample shape contours and another genetic algorithm evolves the appropriate values for the two recognition threshold parameters discussed in Chapter 6. These two genetic algorithms should run concurrently, perhaps on separate processors, and the possibility of unsupervised learning should be investigated.

6. A quantitative measurement of the epistasis of the genes in a chromosome should be further developed, so that it can be confirmed that the use of a genetic algorithm for this type of recognition problem is appropriate. Chromosome maps (Chapter 8) could perhaps be used to advantage in this analysis. Chromosome histogram representations (Chapter 8) should also be further investigated to see if the distribution of the 'best' chromosomes could be used to distinguish differently shaped contours.

Although not directly investigated during the research described in this thesis, the parallel nature of the genetic algorithm can offer speed advantages, especially when implemented directly in hardware. Further research should be considered to investigate the structure of the research suggestions discussed above to find out if the type of genetic algorithm developed in Chapter 5 is suitable for a hardware implementation.

Krishnakumur (1989) explored a small population approach using some very simple genetic parameters. The superior performance in the presence of multi-modality and its merits in solving non-stationary function optimisation problems were demonstrated. This type of genetic algorithm may have advantages when contours in a sequence of images are being analysed. The contour features change with time and will require a rapid response from the genetic algorithm.

Storn (1995) proposed a differential evolution paradigm that can find the correct global minimum, can provide fast convergence, and has a minimum number of control parameters. The method was stated as being inherently parallel. This type of configuration should be investigated to discover if a

number of less fit solutions, rather than just the 'best' solution, could be evolved as required by the genetic algorithm of Chapter 5.

Turton (1994) described the hardware architecture for Very Large Scale Integration (VSLI) of a parallel genetic algorithm. This configuration is reported to be cheap, fast and efficient. This design is of a fine-grained nature where chromosomes are only selected from their nearest neighbours for reproduction. Three types of parallel implementation were reviewed in this paper:

1. A standard implementation in which the fitness evaluation is distributed over a number of processors and the evolution is performed centrally (Fogarty and Huang, 1990).

2. A coarse-grained implementation where several populations evolve in parallel with the 'best' solutions being shared periodically.

3. A fine-grained implementation in which the genetic algorithm acts on each member of the population in parallel. Each chromosome performs crossover and mutation with its neighbours within a defined finite distance.

The coarse-grained implementation should be appropriate for the analysis of multiple populations of the extended chromosome, as suggested in Chapter 8.

Chen (1998) discussed a fine-grained implementation of a parallel genetic simulated annealing Algorithm. A new hybrid algorithm was introduced that, "inherits those aspects of the genetic algorithm that lend themselves to parallelisation, and avoided the serial bottle-necks of the genetic algorithm by incorporating elements of simulated annealing to provide a completely parallel and easily scalable method." The algorithm was reported to scale linearly with the increase of processing elements, "a feature not demonstrated by any previous parallel genetic or simulated annealing algorithm." Additionally the algorithm did not require a careful choice of control parameters.

The reconfigurable nature of Field Programmable Gate Array (FPGA) devices should be investigated to ascertain if any advantages to the genetic algorithm processes investigated in Chapter 5 can be provided. Scott (1995) discussed the use of FPGA hardware designed specifically to implement a genetic algorithm in parallel, and operated as a coprocessor to a PC. The proposed Hardware Genetic Algorithm (HGA) was stated as being 2 to 3 orders of magnitude faster than a software based genetic algorithm. "Due to the re-programmability of the FPGA hardware, the HGA possesses the speed of hardware while retaining the flexibility of a software implementation."

References

I. S. I. Abuhaiba and P. Ahmed, 1993, "A fuzzy graph theoretic approach to recognise the totally unconstrained hand written numerals", Pattern Recognition, 26, 1335-1350.

Y. Abu-Mostafa and D. Psaltis, 1984, "Recognitive aspects of moment invariants", IEEE PAMI, 6, 698-706.

J. N. Amaral et al., 1995, "Designing genetic algorithms for the state assignment problem", IEEE Trans. Sys., Man and Cyb., 25, 687-694.

P. Andry and P. Tarroux, 1994, "Unsupervised image segmentation using a distributed genetic algorithm", Pattern Recognition, 27, 659-673.

G. Anelli et al., 1998, "Decomposition of arbitrarily shaped binary morphological structuring elements using genetic algorithms", IEEE PAMI, 20, 2, 217-224.

H. Asada and M. Brady, 1986, "The curvature primal sketch", IEEE PAMI, 8, 2-14.

G. P. Babu and M. N. Murty, 1993, "A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm", Pattern Recognition Letters, 14, 763-769.

G. P. Babu and M. N. Murty, 1994, "Clustering with evolution strategies", Pattern Recognition, 27, 321-329.

J. Bala and H. Wechler, 1993, "Shape analysis using genetic algorithms", Pattern Recognition Letters, 14, 965-973.

J. Bala and H.Wechler, 1996, "Shape analysis using hybrid learning", Pattern Recognition, 29, 8, 1323-1333.

J. Bala et al., 1997, "Using learning to facilitate the evolution of features for recognising visual concepts", Special Issue of Evolutionary Computation – Evolution Learning and Instinct: 100 years of the Baldwin Effect, MIT Press.

S. Bandyopadhyay and S. K. Pal, 1997, "Pattern classification with genetic algorithms: Incorporation of chromosome differentiation", Pattern Recognition Letters, 18, 119-131.

J. B. Basak et al., 1995, "A connectionist system for learning and recognition of structures: application to handwritten characters", Neural Networks, 8, 643-657.

S. M. Bhandarkar et al., 1994, "An edge detection technique using genetic algorithm-based optimisation", Pattern Recognition, 27, 1159-1180.

B. Bhanu et al., 1995, "Adaptive image segmentation using a genetic algorithm", IEEE Trans. Sys. Man and Cyb., 25, 12, 1543-1567.

A. K. Bhattacharjya and B. Roysam, 1994, "Joint solution of low, intermediate, and high-level vision tasks by evolutionary optimisation: application to computer vision at low SNR", IEEE Trans. Neural Networks, 5, 83-95.

S. Bornholdt and D. Graudenz, 1992, "General asymmetric neural networks and structure design by genetic algorithms", Neural Networks, 5, 327-334.

J. J. Brault and R. Plamondon, 1993, "Segmenting hand written signatures at their perceptually important points", IEEE PAMI, 15, 963-957.

F. Z. Brill et al., 1992, "Fast genetic selection of features for neural network classifiers", IEEE Trans. Neural Networks, 3, 324-328.

R. W. Brockett and P. Maragos, 1994, "Evolution equations for continuous-scale morphological filtering", IEEE Trans. Signal Processing, 42, 3377-3385.

A. M. Bruckstein et al., 1997, "Scale space semi-local invariants", Image and Vision Computing, 15, 335-344.

R. Brunelli and T. Poggio, 1993, "Face recognition: features versus templates", IEEE PAMI, 15, 1042-1052.

K. Brunnstrom and A. Stoddart, 1996, "Genetic algorithms for free-form surface matching", IEEE Proc. ICPR, 689-692.

D. W. Capson, 1984, "An improved algorithm for the sequential extraction of boundaries from a raster scan", Computer Vision, Graphics and Image Processing, 28, 109-125.

G. Cash and M. Hatamain, 1987, "Optical character recognition by the method of moments", Computer Vision, Graphics and Image Processing, 39, 291-310.

R. Cesar and L. Costa, 1996, "Towards effective planar shape representation with multi-scale digital curvature analysis based on signal processing techniques", Pattern Recognition, 29, 9, 1559-1569.

J. Chai and S. Ma, 1998, "Robust epipolar geometry estimation using a genetic algorithm", Pattern Recognition Letters, 19, 829-838.

L. Chambers (Ed.), 1995, "Practical Handbook of Genetic Algorithms - Applications", Vol. 1, CRC Press.

C. C. Chang et al., 1991, "A shape recognition scheme based on relative distances of feature points from the centroid", Pattern Recognition, 24, 1053-1063.

H. Chen et al., 1998, "Parallel genetic simulated annealing: a massively parallel SIMD algorithm", IEEE Trans. Parallel and Distributed Systems, 9, 2, 126-136.

J. Chen et al, 1996, "Segmentation of planar curves into circular arcs and line segments", Image and Vision Computing, 14, 71-83.

J. M. Chen, 1997, "Optimal matching of closed contours with line segments and arcs", Pattern Recognition Letters, 18, 567-574.

K. Chen, 1990, "Efficient parallel algorithms for the computation of two-dimensional image moments", Pattern Recognition, 23, 109-119.

Q. Chen et al., 1994, "Symmetric phase-only matched filtering of Fourier-Mellin transforms for Image registration and recognition", IEEE PAMI, 16, 1156-1168.

Y. Chen et al., 1996, "A parallel genetic algorithm for image restoration", IEEEProc. ICPR, 694-698.

D. Cheng and H. Yan, 1998, "Recognition of hand written digits based on contour information", Pattern recognition, 31, 3, 235-255.

M. Cheriet and C. Y. Suen, 1993, "Extraction of key letters for cursive script recognition", Pattern Recognition Letters, 14, 1009-1017.

D. Chun and H. Yang, 1996, "Robust image segmentation using a genetic algorithm with fuzzy measure", Pattern Recognition, 29, 7, 1195-1211.

L. Cinque and L. Lombardi, 1995, "Shape Description and recognition by a multi-resolution approach", Image and Vision Computing, 13, 599-607.

L. Cinque et al., 1998, "Shape description using cubic polynomial Bezier curves", Pattern Recognition Letters, 19, 821-828.

G. Cong and S. Ma, 1998, "Corner enhancement in curvature scale space", Pattern Recognition, 31, 1491-1501.

P. Cornic, 1997, "Another look at the dominant point detection of digital curves", Pattern Recognition Letters, 18, 13-25.

J. R. Cowell, 1995, "Syntactic pattern recogniser for vehicle identification numbers", Image and Vision Computing, 15, 13-19.

D. Cyganski and R. Vaz, 1995, "A linear signal decomposition approach to affine invariant contour identification", Pattern Recognition, 28, 12, 1845-1853.

S. Das, 1996, "Generic object recognition using multiple representations", Image and Vision Computing, 14, 323-338.

Y. Davidor, 1991, "Genetic Algorithms and Robotics", World Scientific.

L. Davis, 1991, "Handbook of Genetic Algorithms", Van Nostrand Reinhold.

K. De Jong et al., 1997, "Using problem generators to explore the effects of Epistasis", Proc. Seventh International Conference on Genetic Algorithms, July.

A. K. Dewdney, 1989, "Computer Recreations", Scientific American (August), 104-107.

A. Djouadi and E. Bouktache, 1997, "A fast algorithm for the nearest neighbour classifier", IEEE PAMI, 19, 277-282.

M. Dorigo and U.Schnepf, 1993, "Genetics-based machine learning and behaviour-based robotics: a new synthesis", IEEE Trans. Sys. Man and Syb., 23, 141-153.

E. R. Dougherty and C. R. Giardina, 1988, "Mathematical Methods for Artificial Intelligence and Autonomous Systems", Prentice Hall.

T. E. Dufresne and A. P. Dhawan, 1995, "Chord-tangent transformation for object recognition", Pattern Recognition, 28, 1321-1332.

A. Dutta and S. Chaudhury, 1993, "Bengali alpha-numeric character recognition using curvature features", Pattern Recognition, 12, 1757-1770.

T. M. Egan and P. Picton, 1994, "Behaviour of a simple genetic algorithm searching for bright and edge pixels in an image", IEE (London), PG E4 Colloquium, "Genetic Algorithms in Image Processing and Vision", Digest Number 1994/193 (October).

H. Eviatar and R. Somorjai, 1996, "A fast, simple active contour algorithm for biomedical images", Pattern Recognition Letters, 17, 969-974.

P. T. Fairney and D. P. Fairney, 1996, "3D object recognition and orientation from single noisy 2D images", Pattern Recognition Letters, 17, 785-793.

S. Fejes and A. Rosenfeld, 1997, "Discrete active models and applications", Pattern Recognition, 30, 5, 817-835.

J. Flusser and T. Suk, 1993, "Pattern recognition by affine moment invariants", Pattern Recognition, 26, 167-174.

J. Flusser and T. Suk, 1994, "Affine moment invariants: a tool for character recognition", Pattern Recognition Letters, 15, 433-436.

T. C. Fogarty and R. Huang, 1990, "Implementing the genetic algorithm on transputer based parallel processing systems", Parallel Problem Solving in Nature, H. P.Scwefel and R. Manner (Eds), Springer Verlag, New York, 145-149.

D. B. Fogel, 1994, "An introduction to simulated evolutionary optimisation", IEEE Trans. Neural Networks, 5, 3-14.

H. Freeman, 1961, "On the encoding of arbitrary geometric configurations", IRE Trans. Electronic Computers, EC-10, 2, 260-268.

H. Freeman, 1977, "Analysis of line drawings", Digital Image Processing and Analysis,Noordhoff International Publishing, 187-209.

A. M. N. Fu et al., 1997, "A curve bend function based method to characterise contour shapes", Pattern Recognition, 30, 10, 1661-1671.

E. Gelsema, 1996, "Diagnostic reasoning based on a genetic algorithm operating in a Bayesian belief network", Pattern Recognition Letters, 17, 1047-1055.

D. E. Goldberg and J. Richardson, 1987, "Genetic algorithms with sharing for multimodal function optimisation", Proc. Second International Conference on Genetic Algorithms, 41-49.

D. E. Goldberg, 1989, "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley.

D. E. Goldberg, 1990, "Real-coded genetic algorithms, virtual alphabets and blocking", University of Illinois Technical Report.

D. E. Goldberg and K. Deb, 1991, "A comparative analysis of selection schemes used in Genetic Algorithms", in Rawlins (1991, 69-93).

R. C. Gonzalez and P. Wintz, 1987, "Digital Image Processing", Second Edition, Addison Wesley.

G. Granlund, 1972, "Fourier pre-processing for hand print character recognition", IEEE Trans. Computers, 21, 195-201.

J. J. Grefenstette, 1986, "Optimisation of control parameters for genetic algorithms", IEEE Trans. Sys. Man and Cyb., 16, 122-128.

J. Gu and X. Huang, 1994, "Efficient local search with search space smoothing: a case study of the travelling salesman problem (TSP)", IEEE Trans. Sys., Man and Cyb., 24, 728-735.

L. Gupta et al., 1995, "Robust automatic target recognition using a localised boundary representation", Pattern Recognition, 28, 1587-1598.

S. Hata et al., 1996, "Shape extraction of a transparent object using a genetic algorithm",IEEE Proc. ICPR, 684-688.

F. van der Heijden, 1994, "Image Based Measurement Systems", John Wiley.

A. Held et al., 1994, "Towards a hierarchical contour description via dominant point detection", IEEE Trans. Sys. Man and Cyb., 24, 942-949.

T. Hemminger and C. Pomalaz-Raez, 1996, "Rotation and scale independent pattern recognition through optimisation", Pattern Recognition, 29, 3, 487-495.

A. Hill and C. J. Taylor, 1991, "Model-based image interpretation using genetic algorithms", British Machine Vision Conference Report, 266-274.

T. K. Ho et al., 1992, "A word shape analysis approach to lexicon based word recognition", Pattern Recognition Letters, 13, 821-826.

J. Hoffman et al., 1993, "Cluster network for recognition of hand written cursive script characters", Neural Networks, 6, 69-78.

J.H. Holland et al., 1986, "Induction", MIT Press.

J. H. Holland, 1987, "Genetic algorithms and classifier systems; foundations and future directions", Proceedings of Second International Conference on Genetic Algorithms.

J. H. Holland, 1992, "Adaption in Natural and Artificial Systems", Bradford, MIT Press.

A. Hsieh et al, 1996, "Hand written Chinese character recognition by greedy matching with geometric constraint", Image and Vision Computing, 14, 91-104.

J. C. Hsu and S. Y. Hwang, 1997, "A machine learning approach for acquiring descriptive classification rules of shape contours", Pattern Recognition, 30, 2, 245-252.

L. Huang and M. Wang, 1996, "Efficient shape matching through model based shape recognition", Pattern Recognition, 29, 2, 207-215.

X. Huang and J. Gu, 1993, "A constrained approach to multi-font Chinese character recognition", IEEE PAMI, 15, 838-843.

Y. Huang et al., 1995, "Optic flow field segmentation and motion estimation using a robust genetic partitioning algorithm", IEEE, PAMI, 17, 1177-1189.

S. L. Hung and H. Adeli, 1994, "A parallel genetic/neural network learning algorithm for MIMD shared memory machines", IEEE Trans. Neural Networks, 5, 900-909.

T. Hupkens and J. Clippeleir, 1995, "Noise and intensity invariant moments", Pattern Recognition Letters, 16, 371-376.

J. Inesta et al, 1996, "Local symmetries of digital contours from their chain codes", Pattern Recognition, 29, 10, 1737-1749.

J. M. Inesta et al., 1998, "Reliable polygonal approximations of imaged real objects through dominant point detection", Pattern Recognition, 6, 685-697.

H. Ip and W. Wong, 1997, "Fast conditioning algorithm for significant zero curvature detection", IEE Proc. Vis. Image and Signal Processing, 144, 1, 23-30.

H. Ip and D. Shen, 1998, "An affine-invariant active contour model (AI-Snake) for model based segmentation", Image and Vision Computing, 16, 135-146.

C. Jacquelin et al., 1997, "Evolving descriptors for texture segmentation", Pattern Recognition, 30, 1069-1079.

A. Jain and D. Zongker, 1997, "Representation and recognition of hand written digits using deformable templates", IEEE PAMI, 19, 12, 1386-1391.

M. James, 1985, "Classification Algorithms", Collins Professional and Technical Books.

X. Jia and M. Nixon, 1995, "Extending the feature vector for automatic face recognition", IEEE PAMI, 17, 12, 1167-1176.

X. Jiang and H. Bunke, 1991, "Simple and fast computation of moments", Pattern Recognition, 24, 801-806.

A. Jozwik et al., 1998, "A parallel network of modified 1-NN and k-NN classifiers – an application to remote-sensing image classification", Pattern Recognition Letters, 19, 57-62.

G. Kaiser, 1994, "A Friendly Guide to Wavelets", Birkhauser Press.

M. Kass et al., 1988, "Snakes: active contour models", IEEE, Int. JournalComput. Vision, 321-331.

A. Katz and P.Thrift, 1994, "Generating image filters for target recognition by genetic learning", IEEE PAMI, 16, 9, 906-910.

H. Kauppinen et al., 1995, "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification", IEEE PAMI, 17, 201-207.

T. Kawaguchi and M. Nag, 1998, "Recognition of occluded objects by a genetic algorithm", IEEE Proc. ICPR, 233-237.

K. E. Kinnear, 1994, "Advances in Genetic Programming", Bradford, MIT Press.

M. Koch and R. L. Kashyap, 1987, "Using polygons to recognise and locate partially occluded objects", IEEE PAMI, 9, 483-494.

J. Koplowitz and S. Plante, 1995, "Corner detection for chain coded curves", Pattern Recognition, 28, 843-852.

Z. Kovacs, 1995, "A novel architecture for high quality hand printed character recognition", Pattern Recognition, 28, 11, 1685-1692.

J. R. Koza, 1992, "Genetic Programming", Bradford, MIT Press.

J. R. Koza, 1992, "Genetic evolution and co-evolution of computer programs", Artificial Life II, Addison Wesley, 603-629.

J. R. Koza, 1994, "Genetic Programming II", Bradford, MIT Press.

K. Krishnakumar, 1989, "Micro-genetic algorithms for stationary and non-stationary function optimisation", SPIE, Intelligent Control and Adaptive Systems, 1196, 289-296.

L. Kuncheva, 1997, "Fitness functions in editing k-NN reference set by genetic algorithms", Pattern Recognition, 30, 6, 1041-1049.

K. Y. Kupeev and H. J. Wolfson, 1996, " A new method of estimating shape similarity", Pattern Recognition Letters, 17, 873-887.

K. Lai and R. Chin, 1998, "On modelling, extraction, detection and classification of deformable contours from noisy images", Image and Vision Computing, 16, 55-62.

V. F. Leavers, 1992, "Shape Detection in Computer Vision Using the Hough transform", Springer Verlag.

S. Lee and J. C. Pan, 1992, "Off-line tracing and representation of signatures", IEEE Trans. Sys, Man and Cyb., 22, 755-771.

S. W. Lee, 1995, "Multi-layer cluster neural network for totally unconstrained hand written numeral recognition", Neural Networks, 8, 783-792.

S. Y. Lee et al., 1996, "Recognition of human front faces using knowledge-based feature extraction and neuro-fuzzy algorithm", Pattern Recognition, 29, 11, 1863-1876.

B. Li, 1993, "A new computation of geometric moments", Pattern Recognition, 26, 109-113.

B. Li, 1996, "Repeatedly smoothing discrete scale space evolution and dominant point detection", Pattern Recognition, 29, 6, 1049-1059.

R. Lin et al., 1998, "A modified morphological corner detector", Pattern Recognition Letters, 19, 279-286.

J. Lis, 1995, "Genetic algorithm with the dynamic probability of mutation in the classification problem", Pattern Recognition Letters, 16, 1311-1320.

H. C. Liu and M. D. Srinath, 1990, "Partial shape classification using contour matching in distance transformation", IEEE PAMI, 12, 1072-1079.

H. C. Liu and J. S. Huang, 1998, "Pattern recognition using evolution algorithms with fast simulated annealing", Pattern Recognition Letters, 19, 403-413.

R. C. Lo and W. H. Tsai, 1997, "Perspective-transformation-invariant generalised Hough transform for perspective planar shape detection and matching", Pattern Recognition, 30, 3, 383-396.

S. Loncaric and A. P. Dhawan, 1995, "Near-optimal MST-based shape description using a genetic algorithm", Pattern Recognition, 28, 571-579.

B. Lovell and A. Bradley, 1996, "The multi-scale classifier", IEEE PAMI, 18, 2, 124-137.

N. Lu, 1997, "Fractal Imaging", Academic Press.

R. K. Lutz, 1980, "Algorithm for the real time analysis of digitised images", The Computer Journal, 23, 262-268.

S. A. Mahmoud, 1994, "Arabic character recognition using Fourier descriptors and character contour coding", Pattern Recognition, 27, 815-824.

D. Marr and E. Hildreth, 1980, "Theory of edge detection", Proc. Royal Soc. London B, 207, 187-217.

B. Mertzios and K. Tsirikolias, 1993, "Statistical shape discrimination and clustering using an efficient set of moments", Pattern Recognition, 14, 517-522.

Y. Meyer, 1993 "Wavelets, Algorithms and Applications", SIAM Press.

Z. Michalewicz, 1992, "Genetic Algorithms + Data Structures = Evolution Programs",Springer Verlag.

M. Mirmehdi et al., 1997, "Genetic optimisation of the image feature extraction process", Pattern Recognition letters, 18, 355-365.

M. Mitchell, 1996, "An Introduction to Genetic Algorithms", Bradford, MIT Press.

D. A. Mitzias and B. G. Mertzios, 1994, "Shape recognition with a neural classifier based on a fast polygon approximation technique", Pattern Recognition, 27, 627-636.

F. Mokhtarian and A. Mackworth, 1986, "Scale-based description and recognition of planar curves and two-dimensional shapes", IEEE PAMI, 8, 34-43.

F. Mokhtarian, 1995, "Silhouette-based isolated object recognition through curvature scale space", IEEE PAMI, 17, 539-544.

R. Mukundan et al., 1993, "Attitude estimation using moment invariants", Pattern Recognition Letters, 14, 199-205.

C. Murthy and N. Chowdhury, 1996, "In search of optimal clusters using genetic algorithms", Pattern Recognition Letters, 17, 825-832.

F. Neri and L. Saitta, 1996, " Exploring the power of genetic search in learning symbolic classifiers", IEEE PAMI, 18, 11, 1135-1141.

B. Olstad, 1991, "Active contours with grammatical descriptions", British Machine VisionConf. Report, 430-437.

B. Olstad and A. Torp, 1996, "Encoding of a priori information in active contour models", IEEE PAMI, 18, 9, 663-672.

T. Ostrowski, 1995, "Computing with genetic algorithms in the context of adaptive neural filtering", Pattern Recognition Letters, 16, 125-132.

E. Ozcan and C. K. Mohan, 1997, "Partial shape matching using genetic algorithms", Pattern Recognition Letters, 18, 987-992.

S. K. Pal et al., 1994, "Genetic algorithms for optimal image enhancement", Pattern Recognition Letters, 15, 261-271.

D. C. W. Pao and H. F. Li, 1992, "Shape recognition using the straight line Hough transform: theory and generalisation", IEEE PAMI, 14, 1076-1089.

D. C. W. Pao et al., 1997, "An approximate string matching algorithm for on-line Chinese character recognition", Image and Vision Computing, 15, 695-703.

J. M. Pardo et al., 1997, "A snake for model-based segmentation of biomedical images", Pattern Recognition Letters, 18, 1529-1538.

M. Parizeau et al., 1998, "Optimising the cost matrix for approximate string matching using genetic algorithms", Pattern Recognition, 31, 4, 431-440.

J. S. Park and J. H. Han, 1997, "Estimating optical flow by tracking contours", Pattern Recognition Letters, 18, 641-648.

M. J. Paulik et al., 1992, "Non-stationary auto-regressive modelling of object contours", IEEE Trans. Signal Processing, 40, 660-675.

T. Pavlidis, 1980, "Algorithms for shape analysis of contours and waveforms", IEEE PAMI, 2, 301-312.

S. Pei and C. Lin, 1995, "Image normalisation for pattern recognition", Image and Vision Computing, 13, 10, 711-723.

M. Pelillo et al., 1995, "An evolutionary approach to training relaxation labelling processes", Pattern Recognition Letters, 16, 1069-1078.

F. Pernus at al., 1994, "Two-dimensional object recognition using multi-resolution non-information-preserving shape features", Pattern Recognition Letters, 15, 1071-1079.

A. Pikaz and I. Dinstein, 1995, "Matching of partially occluded planar curves", Pattern Recognition, 28, 199-209.

X. Qi and F. Palmieri, 1994, "Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part I: basic properties of selection and mutation", IEEE Trans. Neural Networks, 5, 102-119.

X. Qi and F. Palmieri, 1994, "Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part II: analysis of the diversification role of crossover", IEEE Trans. Neural Networks, 5, 120-129.

G. J. E. Rawlins, 1991, "Foundations of Genetic Algorithms", Morgan Kaufmann.

B. K. Ray and K. S. Ray, 1997, "Scale-space analysis and corner detection on digital curves using a discrete scale-space kernel", Pattern Recognition, 30, 9, 1463-1474.

K. S. Roh and I. S. Kweon, 1998, "2D object recognition using invariant contour descriptor and projective refinement", Pattern Recognition, 31, 4, 441-455.

P. Rosin, 1998, "Determining local natural scales of curves", Patterns Recognition Letters, 19, 63-75.

P. Rosin and S. Venkatesh, 1993, "Extracting natural scales using Fourier descriptors", Pattern Recognition, 9, 1383-1393.

G. Roth and M. D. Levine, 1994, "Geometric primitive extraction using a genetic algorithm", IEEE PAMI, 16, 901-905.

I. Rothe et al., 1996, "The method of normalisation to determine invariants", IEEE PAMI, 18, 4, 366-375.

G. Rudolph, 1994, "Convergence analysis of canonical genetic algorithms", IEEE Trans. Neural Networks, 5, 96-101.

P. Saint-Marc et al., 1993, "B-spline contour representation and symmetry detection", IEEE PAMI, 15, 1191-1197.

H. Saito and M. Mori, 1996, "Object modelling from multiple images using genetic algorithms", IEEE Proc. ICPR, 669-673.

H. Sakano et al., 1996, "Seeing the character images that an OCR system sees - analysis by genetic algorithm", IEEE Proc. ICPR, 411-416.

H. Sardana et al., 1994, "Global description of edge patterns using moments", Pattern Recognition, 27, 109-118.

D. Sarkar, 1993, "A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves", Pattern Recognition Letters, 14, 959-964.

E. Saund, 1990, "Symbolic construction of a 2D scale-space image", IEEE PAMI, 12, 817-830.

P. Scheunders, 1996, "A genetic Lloyd-Max image quantisation algorithm", Pattern Recognition Letters, 17, 547-556.

H. Schiff et al., 1994, "A hardware implementation of a biological neural system for target localisation", IEEE Trans. Neural Networks, 5, 354-362.

S. D. Scott et al., 1995, "HGA: A hardware-based genetic algorithm", Proc. ACM/SIGDA, Third International Symposium on FPGAs, 53-59.

H. Sheu and W. Hu, 1996, "A rotationally invariant two phase scheme for corner detection", Pattern Recognition, 29, 5, 819 - 826.

M. Singh et al., 1997, "Matching structural shape descriptions using genetic algorithms", Pattern Recognition, 30, 9, 1451-1462.

A. Slusek, 1995, "Identification and inspection of 2D objects using new moment-based shape descriptors", Pattern Recognition Letters, 16, 687-897.

J. E. Smith et al., 1994, "Genetic selection of features for clustering and classification", IEE (London), PG E4 Colloquium, "Genetic Algorithms in Image Processing and Vision", Digest Number 1994/193 (October).

M. Sonka et al., 1994, "Image Processing, Analysis and Machine Vision", Chapman and Hall.

W. M. Spears, 1997, "Speciation using tag bits", The Handbook of Evolutionary Computation, T. Baeck et al. (Editors), IOP Publishing and Oxford University Press.

M. Srinivas and L. M. Patnaik, 1994, "Adaptive probabilities of crossover and mutation in genetic algorithms", IEEE Trans. Sys., Man and Cyb., 24, 656-667.

R. Storn and K. Price, 1995, "Differential evolution – a simple and efficient adaptive scheme for global optimisation over continuous spaces", International Computer Science Institute, Technical Report TR-95-012.

E. Tanaka, 1995, "Theoretical aspects of syntactic pattern recognition", Pattern Recognition, 28, 1053-1061.

Y. Tang et al., 1998, "Offline recognition of Chinese handwriting by multi-feature and multi-level classification", IEEE PAMI 20, 5, 556-561.

M. Teague, 1980, "Image analysis via the general theory of moments", J. Opt. Soc. Am., 70, 920-930.

Y. Tian and H. Tsui, 1996, "3D shape recovery from two-colour image sequences using a genetic algorithm", IEEE Proc. ICPR, 674-678.

Q. M. Tieng and W. W. Boles, 1997, "Recognition of 2D object contours using the wavelet transform zero-crossing representation", IEEE PAMI, 19, 8, 910-916.

A. Toet and W. Hajema, 1995, "Genetic contour matching", Pattern Recognition Letters, 16, 849-856.

F. Toyama et al., 1998, "Model-based pose estimation using a genetic algorithm",IEEE Proc. ICPR, 198-201.

O. Trier et al., 1996, "Feature extraction methods for character recognition - a survey", Pattern Recognition, 29, 4, 641-662.

D. M. Tsai and M. F. Chen, 1994, "Curve fitting approach for tangent angle and curvature measurements", Pattern Recognition, 27, 699-711.

P. Tsang, 1997, "A genetic algorithm for invariant recognition of object shapes from broken boundaries", Pattern Recognition Letters, 18, 631-639.

P. W. M. Tsang, 1997, "A genetic algorithm for aligning object shapes", Image and Vision Computing, 15, 819-831.

W. M. Tsang et al., 1994, "Detection of dominant points on an object boundary: a discontinuity approach", Image and Vision Computing, 12, 547-557.

B. C. H. Turton et al., 1994, "A hardware architecture for a parallel genetic algorithm for image registration", IEE (London), PG E4 Colloquium, "Genetic Algorithms in Image Processing and Vision", Digest Number 1994/193 (October).

A. Verri and C. Uras, 1996, "Metric-topological approach to shape representation and recognition", Image and Vision Computing, 14, 189-207.

A. Wallin and O. Kubler, 1995, "Complete sets of complex Zernike moment invariants and the role of the pseudo-invariants", IEEE PAMI, 17, 11, 1106-1110.

H. Wang and M. Brady, 1995, "Real-time corner detection algorithm for motion estimation", Image and Vision Computing, 13, 695-703.

S. Wang et al., 1994, "Invariant pattern recognition by moment Fourier descriptor", Pattern Recognition, 27, 1735-1742.

Y. Wang and N. Funakubo, 1996, "Detection of geometric shapes by the combination of a genetic algorithm and sub-pixel accuracy", IEEE Proc. ICPR, 535-539.

D. Whitley and D. Starkweather, 1990, "Genitor-II: A distributed genetic algorithm", J.Expt. Theor. Artif. Intel., 2, 189-214.

G. R. Wilson, 1997, "Properties of contour codes", IEE Proc. Vis. Image Signal Process., 144, 3, 145-149.

S. W. Wilson, 1985, " Adaptive 'cortical' pattern recognition", Proceedings of First International Conference on Genetic Algorithms and Their Applications, 188-196.

Y. Wong at al., 1998, "Segmented snake for contour detection", Pattern Recognition, 31, 11, 1669-1679.

C. J. Wu and J. S. Huang, 1990, "Human face profile recognition by computer", Pattern Recognition, 23, 255-259.

M. Wu and H. Sheu, 1996, "3D invariant estimation of axi-symmetric objects using Fourier descriptors", Pattern Recognition, 29, 2, 267-280.

M. Wu and H. Sheu, 1997, "Contour-based correspondence using Fourier descriptors", IEE Proc. Vis. Image Signal Process., 144, 3, 150-160.

P. Wunsch and A. F. Laine, 1995, " Wavelet descriptors for multi-resolution recognition of hand printed characters", Pattern Recognition, 28, 1237-1249.

Y. Xin et al., 1996, "Hierarchical contour matching in medical images", Image and Vision Computing, 14, 417- 433.

G. Xu et al., 1994, "Robust active contours with insensitive parameters", Pattern Recognition, 27, 7, 879-884.

G. Yang, 1991, "On the knowledge-based pattern recognition using syntactic approach", Pattern Recognition, 24, 185-193.

L. Yang and R. Prasad, 1993, "On-line recognition of hand written characters using differential angles and structural descriptors", Pattern Recognition Letters, 14, 1019-1024.

L. Yao and W. A. Sethares, 1994, "Non-linear parameter estimation via the genetic algorithm", IEEE Trans. Signal Processing, 42, 927-935.

P. Y. Yin, 1998, "A new method for polynomial approximation using genetic algorithms", Pattern Recognition Letters, 19, 1017-1026.

P. Yip and Y. Pao, 1994, "A guided evolutionary simulated annealing approach to the quadratic assignment problem", IEEE Trans. Sys., Man and Cyb., 24, 1383-1397.

D. Yu and H. Yan, 1997, "An efficient algorithm for smoothing, linearisation and detection of structural feature points of binary image contours", Pattern Recognition, 30, 57-69.

J. Yuan and C. Y. Suen, 1995, "An optimal O(n) algorithm for identifying line segments from a sequence of chain codes", Pattern Recognition, 28, 635-646.

P. C. Yuen et al., 1994, "Robust matching process: a dominant point approach", Pattern Recognition Letters, 15, 1223-1233.

P. C. Yuen et al., 1995, "Localisation of dominant points for image coding", Pattern Recognition Letters, 16, 225-232.

C. T. Zahn and R. Z. Roskies, 1972, "Fourier descriptors for plane closed curves", IEEE Trans. Computers, 21, 269-281.

X. Zhang and D. Zhao, 1997, "A parallel algorithm for detecting dominant points on multiple digital curves", Pattern Recognition, 30, 2, 239-244.

Q. Zhao and T. Higuchi, 1996, "Minimisation of nearest neighbour classifiers based on individual evolutionary algorithms", Pattern Recognition Letters, 17, 125-131.

J. Zhou and T. Pavlidis, 1994, "Discrimination of characters by a multi-stage recognition process", Pattern Recognition, 27, 1539-1549.

P. Zhu and P. M. Chirlian, 1995, "On critical point detection of digital shapes", IEEE PAMI, 17, 737-748.